

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

Олександр ПАВЛОВ

(підпис)

(ініціали, прізвище)

“ ”

2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Програмне забезпечення інформаційних
управляючих систем та технологій»**

спеціальності «121 Інженерія програмного забезпечення»

на тему Мобільне застосування зі зчитування тексту для людей з вадами

зору

Виконав: студент IV курсу, групи

ПІ-63 Лісогор Андрій Юрійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

ст. в. кафедри АСОІУ Халус О.А.

посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові

(підпис)

**Консультант
з графічної
документації**

доц., к.т.н., Ліщук К.І.

посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові

(підпис)

Рецензент:

проф. каф. ТК, д.т.н., проф. Стенін О.А.

посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет(інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – *121 Інженерія програмного забезпечення*

Освітньо-професійна програма – *Програмне забезпечення інформаційних
управляючих систем та технологій*

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис)

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ СТУДЕНТУ**

Лісогору Андрію Юрійовичу.

(прізвище, ім'я, по батькові)

**1. Тема проєкту «Мобільне застосування зі зчитування тексту для
людей з вадами зору»**

керівник проєкту Халус О.А., ст.викладач

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “07” травня 2020 р. №1081-с

2. Термін подання студентом проєкту «08» червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

- 1) *Аналіз вимог до програмного забезпечення: опис предметного середовища, огляд існуючих технічних рішень та відомих програмних продуктів, розробка функціональних та нефункціональних вимог, математичне забезпечення*
- 2) *Моделювання та конструювання програмного забезпечення: моделювання та аналіз програмного забезпечення, архітектура програмного забезпечення, конструювання програмного забезпечення, аналіз безпеки даних*
- 3) *Аналіз якості та тестування програмного забезпечення: аналіз якості ПЗ, опис процесів тестування, опис контрольного прикладу*
- 4) *Впровадження та супровід програмного забезпечення: розгортання програмного забезпечення, робота з програмним забезпеченням*

5) Перелік графічного матеріалу

1) Схема структурна варіантів використання

2) Схема структурна класів програмного забезпечення

3) Креслення вигляду екранних форм

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «10» березня 2020 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	19.03.2020	
2.	Аналіз існуючих методів розв'язання задачі	26.03.2020	
3.	Постановка та формалізація задачі	26.03.2020	
4.	Аналіз вимог до програмного забезпечення	02.04.2020	
5.	Алгоритмізація задачі	02.04.2020	
6.	Моделювання програмного забезпечення	09.04.2020	
7.	Обґрунтування використовуваних технічних засобів	16.04.2020	
8.	Розробка архітектури програмного забезпечення	23.04.2020	
9.	Розробка програмного забезпечення	30.04.2020	
10.	Налагодження програми	07.05.2020	
11.	Виконання графічних документів	14.05.2020	
12.	Оформлення пояснювальної записки	21.05.2020	
13.	Подання ДП на попередній захист	28.05.2020	
14.	Подання ДП рецензенту	03.05.2020	
15.	Подання ДП на основний захист	08.06.2020	

Студент _____ Андрій ЛІСОГОР
(підпис)

Керівник проєкту _____ Олена ХАЛУС
(підпис)

[illegible]

АНОТАЦІЯ

Пояснювальна записка складається з 4 розділів, містить 18 рисунків, 17 таблиць, 13 джерел, що в сумі складає 63 сторінки.

Об’єкт дослідження: Мобільне застосування зі зчитування тексту для людей з вадами зору.

Мета дипломного проєкту: допомогти людям з порушеннями зору отримувати та сприймати текстову інформацію, за допомогою зручного IOS додатку з підтримкою різноманітних звукових індикацій.

В першому розділі даної роботи було досліджено предметну область пов’язану з алгоритмами OCR та системами OCR для створення мобільного IOS додатку. Зазначено відомі аналоги, описано їх переваги та недоліки.

В другому розділі описно моделювання та конструювання додатку. Була розроблена архітектура застосунку, описані програмні продукти, які були використані.

В третьому розділі було складено тестовий план, визначено об’єкти тестування, визначено підходи до тестування та описано основні задачі тестування.

В четвертому розділі наведено опис впровадження мобільного додатку та представлено керівництво користувача.

КЛЮЧОВІ СЛОВА: *МОБІЛЬНЕ ЗАСТОСУВАННЯ, CORE ML, ЗЧИТУВАННЯ ТЕКСТІВ, КОМП’ЮТЕРНИЙ ЗІР.*

					КПІ.ІП-6317.045490.02.81	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

The explanatory note of the diploma project consists of four sections, 18 pictures, 17 tables, 13 sources – in total 63 pages.

The object of study: Mobile scanning app for visually impaired people.

The aim of diploma project: To help visually impaired people to receive and perceive textual information, using a convenient iOS application that supports a variety of audio indications.

In the first section of this paper, the subject area related to OCR algorithms and OCR systems for creating a mobile IOS application was explored. Known analogues are mentioned, their advantages and disadvantages are described.

The second section describes the modeling and design of the application. An application architecture was developed, described the software products that were used. In the third section, a test plan was drawn up, the objects of testing were identified, approaches to testing were defined, and the main tasks of testing were described.

The fourth section describes the implementation of the mobile application and provides a user guide.

KEYWORDS: *MOBILE APPLICATION, CORE ML, TEXT SCANNING, OCR.*

Пояснювальна записка до дипломного проєкту

на тему: Мобільне застосування зі зчитування тексту для людей з вадами зору

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	10
ВСТУП.....	11
РОЗДІЛ 1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
1.1 Загальні положення.....	13
1.2 Змістовний опис і аналіз предметної області	14
1.2.1 Сканування та сегментація.....	15
1.2.2 Попередня обробка.....	15
1.2.3 Розпізнання.....	16
1.2.4 Пост-обробка та зчитування тексту.....	16
1.3 Аналіз успішних ІТ-проектів.....	17
1.3.1 Аналіз відомих технічних рішень.....	17
1.3.2 Аналіз відомих програмних продуктів.....	20
1.4 Аналіз вимог до програмного забезпечення.....	24
1.4.1 Розроблення функціональних вимог.....	24
1.4.2 Розроблення нефункціональних вимог.....	32
1.4.3 Постановка комплексу завдань модулю.....	33
1.5 Математичне забезпечення.....	33
1.6 Висновки до розділу.....	36
РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО	
ЗАБЕЗПЕЧЕННЯ.....	37
2.1 Моделювання та аналіз програмного забезпечення.....	37
2.2 Архітектура програмного забезпечення.....	39
2.2.1 Зберігання даних.....	39

2.3	Архітектурний патерн.....	49
2.4	Висновок до розділу.....	50

РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО

ПРОДУКТУ.....51

3.1	Тестовий план	51
3.1.1	Вступ.....	51
3.1.2	Об'єкти тестування	51
3.1.3	Функції що не обов'язково будуть тестуватися.....	52
3.1.4	Функції що обов'язково будуть тестуватися.....	52
3.1.5	Підхід.....	53
3.1.6	Критерій пройдений/невдалий	53
3.1.7	Результат тестування.....	53
3.1.8	Задачі тестування.....	54
3.1.9	Вимоги до середовища	54
3.1.10	Відповідальність.....	54
3.1.11	Затвердження.....	54
3.2	Опис контрольного прикладу.....	55
3.3	Висновок до розділу.....	59

РОЗДІЛ 4 АНАЛІЗ ТА СУПРОВІД ПРОГРАМНОГО

ЗАБЕЗПЕЧЕННЯ.....60

4.1	Розгортання програмного забезпечення.....	60
4.2	Робота з програмним забезпеченням	60
4.3	Висновки до розділу	60

ВИСНОВКИ.....61

ПЕРЕЛІК ПОСИЛАНЬ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ	програмне забезпечення
UI (User Interface)	інтерфейс користувача
Framework	(каркас, платформа, структура, інфраструктура) – інфраструктура програмних рішень, що полегшує розробку складних систем
IDE (Integrated Development Environment)	комп'ютерна програма, що допомагає програмістові розробляти нове програмне забезпечення чи модифікувати (удосконалювати) вже існуюче
CNN (Convolutional Neural Net)	Розгорткова нейронна мережа
OCR (Optical Character Recognition)	оптичне розпізнавання символів
Vision Framework	фреймворк від компанії Apple, що дозволяє впровадити алгоритми комп'ютерного зору до свого додатку
CoreML	фреймворк від компанії Apple, що дозволяє інтегрувати навчену модель до свого застосунку

ВСТУП

В наш час мобільні пристрої відчиняють для людини велику кількість можливостей. Ми слухаємо музику, займаємося серфінгом в інтернеті, проводимо дозвілля за іграми, дивимося кіно тощо. Але головним є те, що розвиток мобільних технологій дозволяє людям з різноманітними вадами компенсувати їх. Так сліпий чоловік може орієнтуватися в просторі навіть без собаки – поводиря, використовуючи застосунок, який аналізує навколишнє середовище та описує його користувачу. Також, може користуватися онлайн-банкми, визначати номінали купюр, дивитися кіно з коментарями. Все це доступне завдяки таким функціям як VoiceOver (iOS) і TalkBack (Android). Але, досі є речі, котрі обійшла стороною увага розробника. Одним з важливих та невирішених питань є мобільний застосунок, яким могла б користуватися людина з вадами зору та який дозволяв би їй сприймати текстову інформацію.

Компанія Apple надає широкий спектр технологій для розробника, що дозволяють йому впровадити алгоритми комп'ютерного зору в свій проєкт (Vision framework) та полегшити класифікацію об'єктів (Core ML) [1].

Метою розробки є створення застосунку, який дозволяє зчитування тексту використовуючи камеру смартфона, надає звуковий супровід основних етапів під час використання застосунку (Наведення на текст, розпізнавання, завершення розпізнавання тощо.), аналізує отриману інформацію та зачитує її.

Завданням даної роботи є розробка додатку для ОС iOS.

Для успішного виконання даної дипломної роботи необхідно:

- оглянути наукову літературу з обраної теми;
- провести огляд архітектури, інструментів програмування, та ключових моментів при розробці мобільних додатків;
- створити прототип майбутньої програми;
- розробити додаток згідно створеного прототипу;
- розробити інтерфейс користувача.

Пояснювальна записка складається з чотирьох розділів.

Перший розділ складається із аналізу загальних положень, змістовного опису предметної області, дослідження існуючих аналогів на ринку . Цей розділ включає також опис програмних продуктів, які допомогли реалізувати завдання розробки.

Другий розділ містить опис архітектури та моделювання програмного забезпечення.

Третій розділ описує створення детального тест плану. Також в ньому містяться результати проведених тестів.

В четвертому розділі наведено опис впровадження мобільного додатку та представлено керівництво користувача.

Публікації:

– Лісогор А.Ю. МОБІЛЬНЕ ЗАСТОСУВАННЯ ЗІ СЧИТУВАННЯ ТЕКСТУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ// IV Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020».

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Мобільні пристрої в наш час кардинально змінили життя людей з порушеннями зору. Завдяки різноманітним голосовим асистентам, функціям проговорення тексту - така людина отримала змогу користуватися майже усіма програмами без яких уявити повсякденне життя вже майже неможливо. Води зору більше не стають на заваді використанню месенджерів, систем онлайн – банкінгу, навігаторів, браузерів тощо. І хоча наш світ впевнено крокує в повну автоматизацію та цифровізацію, не всі проблеми до цього часу знайшли свій розв’язок. Уявіть, що людина з певними розладами в зоровому апараті хоче прочитати лист, цінник в супермаркеті, об’яву тощо. Це може стати для неї великою перешкодою. На допомогу приходить мобільний телефон з камерою. Компанія Apple надає широкий спектр технологій для розробника, що дозволяють йому впровадити алгоритми комп’ютерного зору (OCR) в свій проєкт (Vision framework) та полегшити класифікацію об’єктів (Core ML) тим самим створивши систему оптичного розпізнавання.

Системи оптичного розпізнавання це один з небагатьох інструментів, що мають змогу надати людям з вадами зору можливість отримати доступ до печатної інформації. Загалом така система складається з трьох основних частин:

- сканування;
- оптичне розпізнавання символів (англ. OCR);
- зачитування тексту за допомогою синтезованої мови.

Оптичне розпізнавання символів (англ. Optical character recognition) – це електронна трансляція зображень рукописного, машинописного або друкованого тексту в текстові дані.

Точне розпізнавання символів в печатному тексті можливе лише якщо наявні чіткі зображення. Точність в такому випадку може перевищувати 99%. Це досягається сегментацією зображення, обробкою та подальшою класифікацією. Ще однією з широко досліджувальних тем є - розпізнавання рукописного тексту. Але в даний час точність сильно нижча ніж для “печатного” тексту. Форми окремих рукописних символів іноді можуть не містити достатньої кількості інформації для того, щоб розпізнати весь текст.

Отже, було прийнято рішення створити мобільний застосунок, що об’єднає зазначені технології та вирішить проблему сприйняття текстової інформації для людей з вадами зору.

1.2 Змістовний опис і аналіз предметної області

Створений мобільний застосунок містить реалізацію OCR-системи, спроектованої для вирішення специфічної задачі - зчитування, аналізу та зачитування тексту (Рисунок 1.1) [1].



Рисунок 1.1 – Схема OCR-системи

1.2.1 Сканування та сегментація

Перший компонент в OCR-системі це оптичне сканування. Під час сканування ми захоплюємо цифрове зображення нашого документа. Наступний компонент – це сегментація. Вона визначає складові зображення, знаходить регіон документа. Більшість OCR алгоритмів поділяють слова на відокремлені літери, котрі в подальшому розпізнаються індивідуально. Основними проблемами під час сегментації є:

- відокремлення поєднаних літер;
- відокремлення тексту від інших “шумів”;
- неправильне трактування графіки та геометрії з текстом і навпаки.

1.2.2 Попередня обробка

Третім компонентом OCR-системи є попередня обробка (англ. Pre-processing). На цьому рівні дані піддаються ряду попередніх етапів обробки, що робить їх придатними для подальшого розпізнання. Це робиться через те, що зображення після сканування може містити певну кількість “шуму” чи бути розмитими.

Деякі з цих дефектів можуть спричинити поганий рівень розпізнавання (знижити точність). Наприклад, під час початкової обробки заповнюються невеликі розриви, прогалини і дірки в оцифрованих символах, витончення зменшує ширину ліній тощо. Також, процес даної обробки включає нормалізацію зображень (Рисунок 1.2) [2].



Рисунок 1.2 - Приклад нормалізації

1.2.3 Розпізнання

Четвертий компонент це розпізнання. На даному етапі ми передаємо оброблене зображення до CNN моделі, яка аналізує та класифікує його. Згорткова нейронна мережа (англ. CNN) – це клас глибоких штучних нейронних мереж, який застосовують для аналізу зображень (Рисунок 1.3)[5].

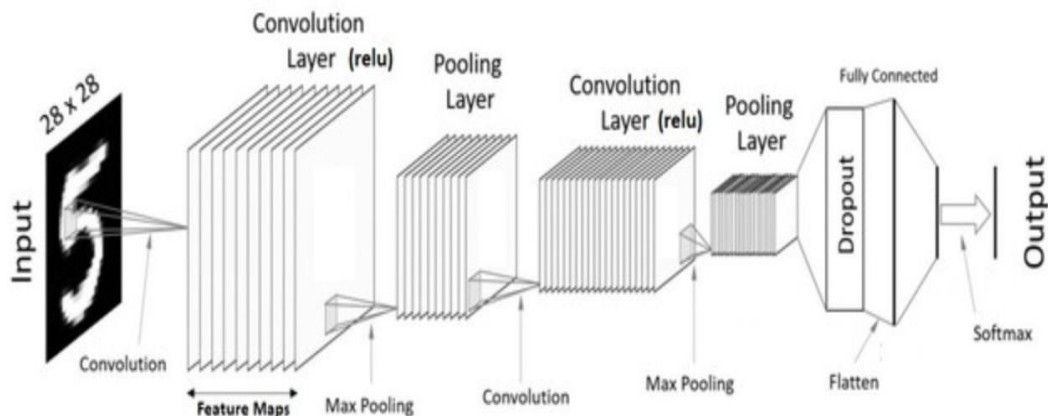


Рисунок 1.3 - Схема роботи CNN мережі

1.2.4 Пост-обробка та зачитування тексту

На даному етапі виконується утворення слів і подальше їх поєднання в текст. Робиться це через те, що результатом розпізнавання зображення є індивідуальні символи. Групування символів базується на їх розташуванні в документі.

Важливим кроком є реалізація зачитування новоутвореного тексту. Такі компанії як Apple надають розробникам великий інструментарій для впровадження синтезованого проговорення тексту в проєкті. Для мови програмування Swift наявні такі класи: `AVSpeechSynthesizer`, `AVSpeechUtterance` та `AVSpeechSynthesisVoice`, які доступні в бібліотеці `AVFoundation`.

1.3 Аналіз успішних ІТ-проектів

1.3.1 Аналіз відомих технічних рішень

Створений мобільний застосунок можна умовно поділити на 3 частини. Перша – це OCR-система, яка виконує основний функціонал зі зчитування та аналізу тексту. Друга – реалізація зберігання розпізнаних текстів. І на решті третя частина - зручний UI з підтримкою VoiceOver. Роздивимось кожен з частин детальніше.

Так як додаток розроблено для платформи IOS, це відкриває широкий спектр технічних рішень. Для реалізації OCR-системи були використані такі програмні продукти як:

- visionframework;
- coreML;
- avspeechsynthesizer, avspeechutterance, avspeechsynthesisvoice.

Vision framework - фреймворк розроблений компанією Apple, що надає можливість задіяти високо продуктивний аналіз зображень та технологію комп'ютерного зору до зображень та відео. Дозволяє виявляти штрих-коди, текст, об'єкти і тд.

В Vision наявні 3 основні категорії класів: VNRequest, VNSequenceRequestHandler, VNObservation. VNRequest - цей абстрактний суперклас описує запит користувача до Vision (VNDetectTextRectanglesRequest, VNDetectBarcodesRequest та інші). VNSequenceRequestHandler - виконує бажану кількість реквестів для зображення. VNObservation - абстрактний суперклас для результатів аналізу зображення[6] (Рисунок 1.4).

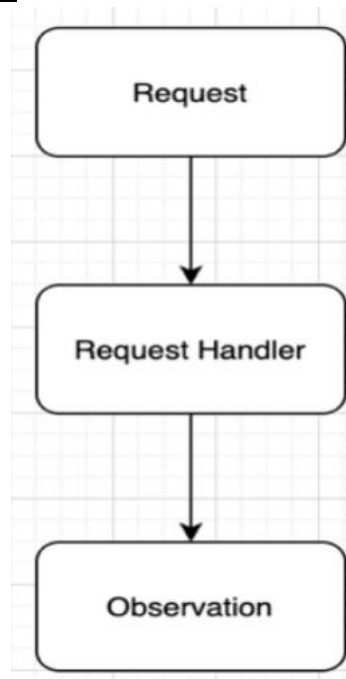


Рисунок 1.4 –Схема механізму роботи Vision

CoreML-декілька років тому компанія Apple представила цей фреймворк, який сильно полегшує роботу з технологіями машинного навчання. В його основі лежить ідея про те, що можна взяти преднавчену модель даних та легко інтегрувати її в свій застосунок. Core ML використовує низькорівневі Metal, Accelerate, BNNS і тому результати обчислень надходять дуже швидко. Ядро підтримує нейронні мережі, generalized linear models, feature engineering, tree ensembles, pipeline models[7].

Окремо необхідно зазначити програмні продукти, котрі використовуються для тренування CNN-моделі та її подальшого трансформування в формат .mlmodel, необхідний для застосування з фреймворком CoreML. Це такі відомі рішення як:

- keras;
- coremltools.

Keras - відкрита нейромережна бібліотека, що створена на мові Python, що дозволяє створювати моделі глибокого навчання (Рисунок 1.5). Keras працює за такими принципами [8]:

- модульність – модель можна розуміти як послідовність або графік;
- мінімалізм – бібліотека забезпечує достатньо для досягнення результату, без надмірностей;
- розширюваність – нові компоненти легко додавати, дослідження нових ідей - можливе.

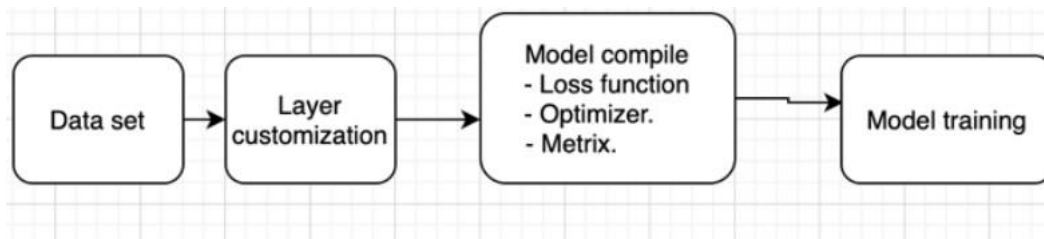


Рисунок 1.5 - Схема роботи Keras

Coremltools – це пакет python для створення, вивчення та тестування моделей у форматі .mlmodel [4]. Зокрема, його можна використовувати для:

- перетворювати підготовлені моделі в формат .mlmodel;
- перевіряти конверсію використовуючи CoreML.

Отже, загальна схема створення та підготовки моделі для використання з CoreML поєднує все вищеперелічені етапи (Рисунок 1.6).

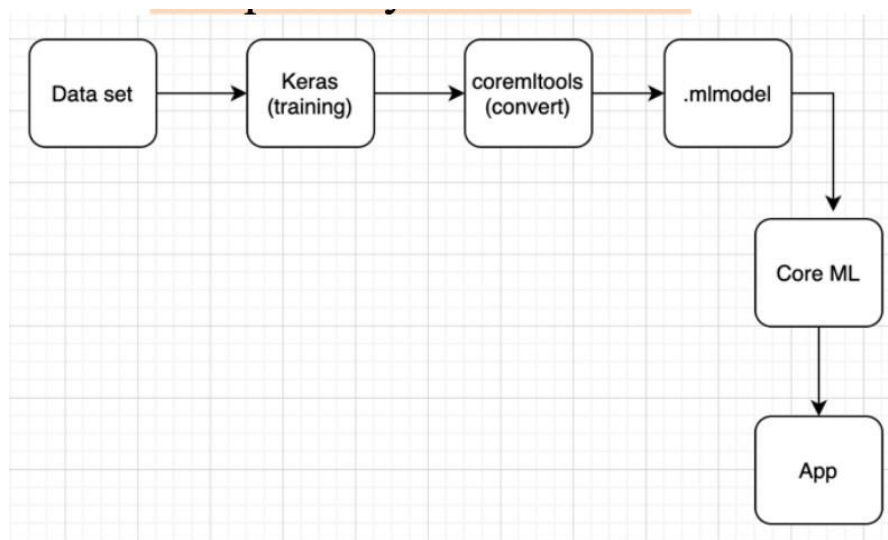


Рисунок 1.6. - Схема створення та трансформації моделі

Для зберігання розпізнаних текстів було використано Core Data. Core Data - це фреймворк, який керує та зберігає дані в застосунку. Можна дивитись на цей фреймворк, як на оболочку над фізичним реляційним сховищем. Core Data використовує 4 типи сховищ:

- sqllite;
- binary;
- in-memory;
- xml (тільки для MacOS).

Задля того, щоб створити умови в застосунку для людей з вадами зору, було виконано налаштування Accessibility для головних UI елементів в коді програми.

1.3.2 Аналіз відомих програмних продуктів

Один з найвідоміших мобільних застосунків з вищезазначеної предметної області – це Pen To Print (Рисунок 1.7) [3]. Це хороший додаток з рейтингом на AppStore 4.4/5.

Створений компанією Serendi LTD, він зчитує та аналізує рукописний текст, надає можливість зберігати розпізнаний текст [3]. Але, в даному рішенні наявні також і мінуси. Погано налаштований Accessibility, що унеможлиблює користування людьми з вадами зору. Немає також підтримки зачитування збереженого тексту.

Повний функціонал надається користувачу тільки після оформлення підписки. Протестувавши даних застосунк, я зробив деякі висновки. Серед них:

- коректність розпізнавання рукописного тексту потребує дуже чіткого відокремленого стилю письма;
- відсутність автоматичного розпізнавання листів. Тобто людина має власноруч вибрати регіон для зчитування.



Рисунок 1.7 - Pen to Print логотип

Інший відомий застосунок – TextScanner (OCR)

(Рисунок

1.10)[12]. Має рейтинг 4.7/5 на App Store. Серед плюсів можна зазначити підтримку 92 мов для сканування та розпізнавання, можливість перекласти відсканований текст. Результат сканування можна зберігати. Серед мінусів:

- можливість зчитування лише із збережених фотографій;
- відсутня підтримка зачитування текстів.
- погано налаштований Accessibility;
- повний функціонал надається лише після купівлі підписки.



Рисунок 1.8 - Pen-to-print приклад роботи

					КП.ІП-6317.045490.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Протестувавши даний застосунок, я виявив, що він стикається з тією ж проблемою що і Pen to Print – досить низька точність розпізнання.

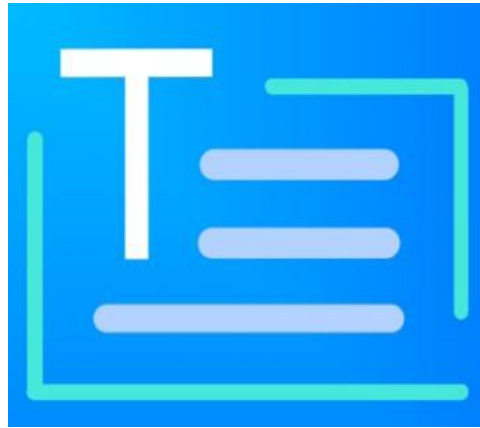


Рисунок 1.9 - Text Scanner (OCR) логотип

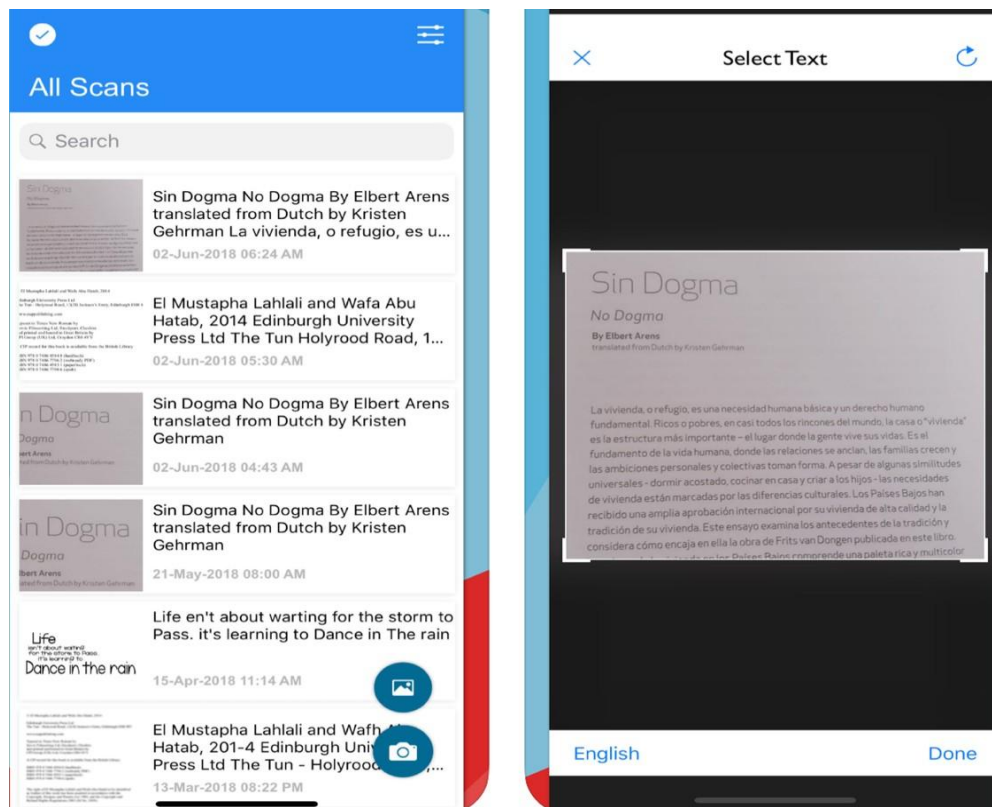


Рисунок 1.10 - Приклади роботи Text Scanner (OCR)

FineScanner AI PRO-сканер OCR від компанії ABBYY, знаходиться в рейтингу “Бізнес” на 28 місці в App Store з оцінкою 4.7/5 (Рисунок 1.11) [13]. Цей застосунок використовує штучний інтелект для створення копій документів та книг в форматах PDF та JPEG. Головні можливості:

- сканує печатні та рукописні бумажі та зберігає їх в JPEG таPDF;
- голосові команди Siri, що дозволяє відкриватискани;
- самостійно віднаходить границі сканованоготексту;
- зберігання відсканованих текстів надевайсі.

Середмінусівможназначитите,щонавітьпринаявностіпідтримкиголосових команд Siri, використання для людей з вадами зору залишається незручним, бо більша частина UI буде незадіяна ними. Незважаючи на заявлену точність розпізнання тексту на практиці виконується менше ніж в 60%. Також слід звернути увагу, що це найдорожчий застосунок із сімейства схожих з ним.



Рисунок 1.11 - Логотип FineScanner AI PRO

1.4 Аналіз вимог до програмного забезпечення

Цільовий користувач даного програмного забезпечення - це людина з вадами зору, тому дуже важливо, щоб користувацький інтерфейс був зручним, інтуїтивно зрозумілим. Основні етапи використання даного додатку мають супроводжуватися голосовими, звуковими позначеннями.

1.4.1 Розроблення функціональних вимог

Мобільний застосунок повинен відповідати таким функціональним вимогам:

- зчитування та розпізнавання тексту в режимі камери;
- автовизначення координат тексту;
- відображення локацій текстів;
- збереження тексту;
- зачитування тексту;
- видалення текстів;
- ввімкнення та вимкнення додаткових звукових позначень.

Для більш детального дослідження вимог висунутих до мобільного додатку прослідкуємо за можливими випадками використання:

Таблиця 1.1 – Варіант використання UC001

Назва	Зчитування тексту в режимі камери
Опис	Користувач розпочинає сканування, після чого при вдалому скануванні лунає сигнал

Продовження таблиці 1.1

Учасники	Користувач
Післяумови	Текст зчитано та розпізнано
Основний сценарій	Користувач тисне на кнопку “Scan” і чує синтезоване слово “Scan” , після цього відкривається камера. Користувач наводить камеру на текст і чекає поки пролунає звукове позначення. Після чого текст розпізнано.

Таблиця 1.2 – Варіант використання UC002

Назва	Автовизначення кордону тексту
Опис	Коли користувач знаходиться в режимі камери, у випадку якщо в об’єктив попадає текст - його границі визначаються автоматично
Учасники	Користувач

Продовження таблиці 1.2

Передумови	Користувач знаходиться в режимі сканування
Післяумови	Границі тексту визначено
Основний сценарій	Після переходу в режим сканування та наведення на текстовий блок, границі цього текстового блоку визначаються автоматично

Таблиця 1.3 – Варіант використання UC003

Назва	Збереження тексту
Опис	Після зчитування текст зберігається в бібліотеку користувача
Учасники	Користувач
Передумови	Користувач має відсканувати текст
Післяумови	Текст збережено в бібліотеці користувача

Продовження таблиці 1.3

Основний сценарій	Після сканування тексту, користувач може натиснути на кнопку “Collection” та перейти до переліку всіх збережених текстів.
-------------------	---

Таблиця 1.4 – Варіант використання UC004

Назва	Зачитування тексту
Опис	Користувач після натискання на конкретний текст, має змогу натиснути на кнопку “Listen” та вибраний текст проговориться синтезованою мовою
Учасники	Користувач
Передумови	Наявність збереженого тексту
Післяумови	Текст проговорівся синтезованою мовою
Основний сценарій	Після вибору тексту та на натискання на рядок з ним відбувається перехід на новий екран де користувач має змогу прослухати текст натиснувши на кнопку “Listen”

Таблиця 1.5 – Варіант використання UC005

Назва	Видалення текстів
Опис	Користувач після натискання на конкретний текст, має змогу натиснути на клавішу “Delete” та вибраний текст видалиться з бібліотеки
Учасники	Користувач
Передумови	Наявність збереженого тексту
Післяумови	Текст видалено з бібліотеки
Основний сценарій	Після вибору тексту та на натискання на View з ним відбувається перехід на новий екран де користувач має змогу видалити текст натиснувши на кнопку “Delete”

Таблиця 1.6 – Варіант використання UC006

Назва	Позначення локацій текстів на мапі
Опис	Перейшовши на екран мапи, користувач має можливість переглянути локації зчитаних текстів
Учасники	Користувач

Продовження таблиці 1.6

Передумови	Наявні збережені тексти в бібліотеці
Післяумови	Відображення локації користувача та локацій відображених текстів
Основний сценарій	Користувач в меню вибору опції натискає на кнопку Map, далі перейшовши на вищезазначений екран він бачить мапу з відслідкованою на ній власну локацію та позначки з текстами

Таблиця 1.7 – Варіант використання UC007

Назва	Ввімкнення та вимкнення додаткових звукових позначень
Опис	Перейшовши на екран налаштувань, користувач має змогу ввімкнути та вимкнути додаткові звукові позначення, натиснувши на кнопку On/Off
Учасники	Користувач

Продовження таблиці 1.7

Передумови	Відсутні
Післяумови	В залежності від попереднього стану, додаткові позначення ввімкнені або вимкнені.
Основний сценарій	Користувач в меню вибору опції натискає на кнопку з позначкою ключа, та переходить на екран налаштувань, де натиснувши на кнопку on/off, в залежності від попереднього стану, додаткові позначення вмикаються або вимикаються.

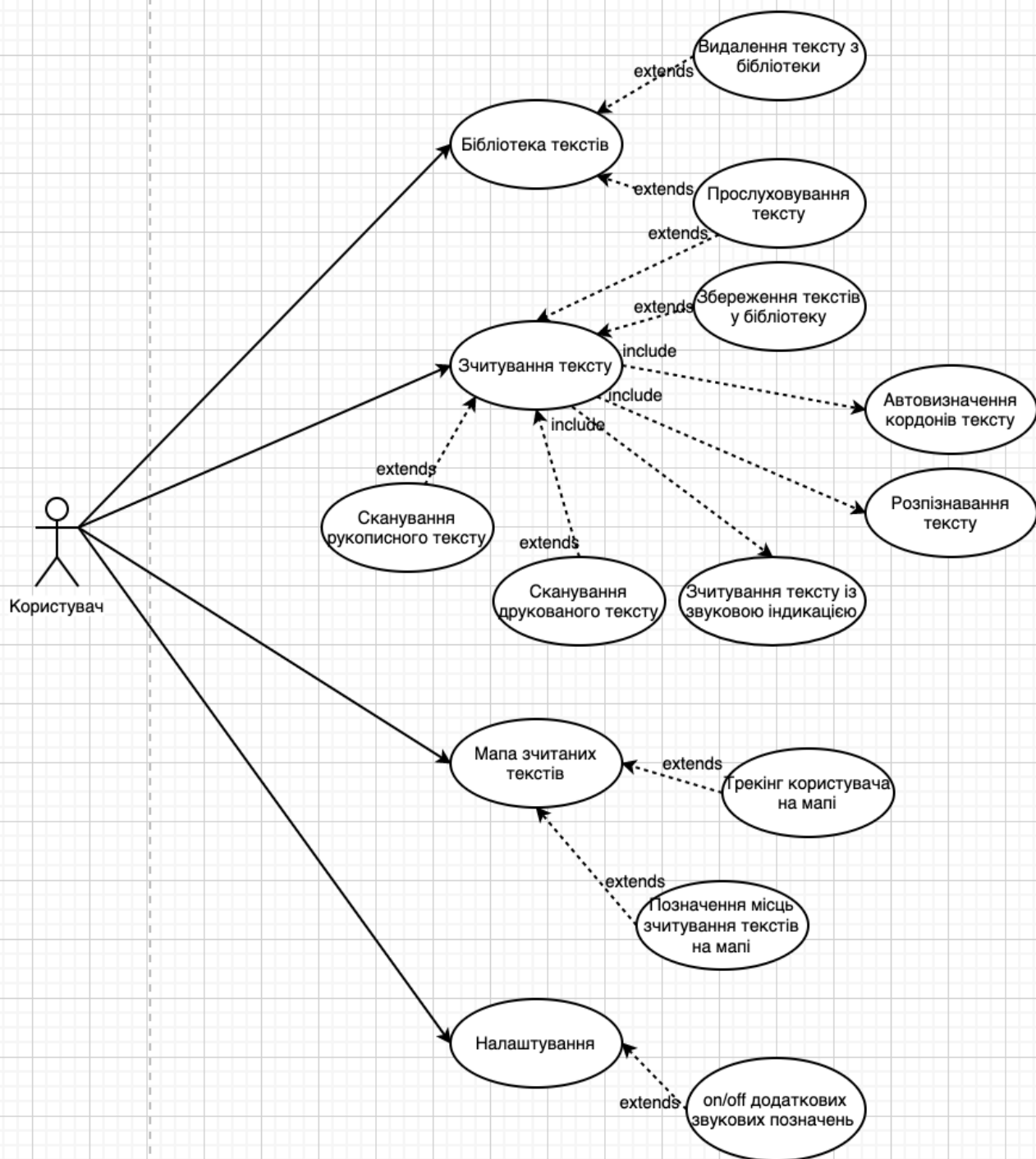


Рисунок 1.12 - Схема структурна варіантів використання

Таблиця 1.8 – Матриця залежності між вимогами і варіантами використання

	Зчитування та розпізнавання тексту	Автовизначення кордону тексту	Збереження тексту	Видалення тексту	Зчитування тексту	Позначення локацій текстів на мапі	On/Off Додаткових позначень
UC001							
UC002							
UC003							
UC004							
UC005							
UC006							
UC007							

Зобразимо матрицю трасування для того, щоб перевірити чи покривають варіанти використання всі функціональні вимоги (Таблиця 1.8).

1.4.2 Розроблення нефункціональних вимог

Мобільний додаток має відповідати таких нефункціональним вимогам:

- має працювати на пристроях з версією IOS 13 та вище;
- повинно мати простий та інтуїтивно зрозумілий інтерфейс.

1.4.3 Постановка комплексу завдань модулю

Мобільне застосування має вирішувати задачі отримання, сприймання текстової інформації людьми з певними вадами зору.

Метою написання даного додатку – створення програмного засобу, що допоможе людям з розладами зорового апарату сприймати звичайну текстову інформацію.

Задля того, щоб досягти мети додатком мають вирішуватися такі задачі:

- сканування рукописного та печатного текстів та їх подальше зачитування;
- можливість збереження текстів до па'мяті пристрою;
- відслідковування позицій зчитуваних текстів та користувача на мапі;
- звуковий супровід основних етапів використання застосування.

Програма має підтримуватися пристроями як “iphone” та “ipodtouch” з версією IOS не нижче 13.

1.5 Математичне забезпечення

Для того, щоб виконувати класифікацію, розпізнавання символів була створена CNN (Convolutional Neural Network) (Рисунок 1.13) [5]. Для створення моделі було використано доповнений MNIST dataset відомий як EMNIST. Цей dataset включає 814255 семплів, 62 класи : числа 0-9 та A-Z як великих так і малих. Результати які показує дана модель після тренування - досить оптимістичні. Точність даної моделі складає 86%. Але, було досліджено, що після трансформування даної моделі в формат .mlmodel, який необхідний для роботи з фреймворком CoreML, точність знизилась. Для покращення результатів класифікації було прийнято рішення застосувати до створених слів AutoComplete. Ця модель приймає на вхід нормалізовані, grayscaled зображення розміром 28x28. Обробку зображень для подальшого використання з .mlmodel виконує Vision Framework.

Від розробника вимагається лише вказати параметри (розміри орієнтація ітд.).

Як можна побачити тренувана модель містить декілька convolutional слоїв (Рисунок 1.13). Для розуміння того, яка працює даний прошарок уявимо матрицю розміром 6×6 , параметрами якої є певні значення (так як у зображення у нас grayscale то ці значення будуть від 0-255). Далі ініціалізуємо вагову матрию

3×3 , яка витягує певні особливості із зображень. Ця вага “перебігає” зображення таким чином, щоб усі пікселі були покриті хоча б один раз, для отримання згорнутого виходу. Після цих “перебігань”, паралельно з якими виконується додавання значень, які отримані від множення виділеної 3×3 матриці на вагову матрицю, ми отримуємо зображення 4×4 .

Вагова матриця поводить ся як фільтр для зображення, витягуючи конкретну інформацію з вихідної матриці. Так як в нашому випадку наявні декілька згорткових шарів, витягується більша кількість особливостей зображення, мережа стає глибшою і вірогідність коректного спрацювання моделі збільшується.

Також, дивлячись на представлену модель ми можемо помітити, що після кожного згорткового прошарку та Dense прошарку було додано BatchNormalization layer. Зазвичай модель повинна бути досить складною для того, щоб отримати значне покращення від застосування batchnormalization. Даний тип нормалізації дозволяє пришвидшити тренування моделі. Дуже важливою частиною CNN є таке поняття як підвиборка. MaxPooling2D зменшує розміри зображення шляхом додавання значень блоків пікселів. До матриці нового зображення переноситься максимальне число наявних діагоналей певним шагом. Що в результаті дає зменшене зображення. Наявні також такі прошарки як flatten, який з двумірної масиви перетворює зображення на одновірний масив завдяки вистраюванню рядків пікселів в один ряд. Є також два Dense прошарки, що представляють з себе пов'язані нейронні слої. Кожний вузол вказує на вірогідність відношення кожного конкретного зображення до одного з класів зображень.

Змн.	Арк.	№ докум.	Підпис	Дата	
------	------	----------	--------	------	--

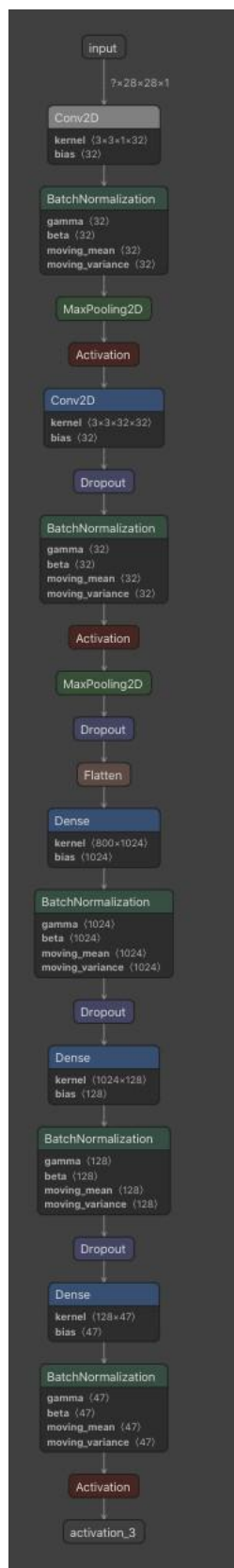


Рисунок 1.13- Схема тренованої CNN моделі

1.6 Висновки до розділу

В даному розділі були описані функціональні та нефункціональні вимоги. На основі порівняння функціональних вимог та можливих сценаріїв використання була створена матриця залежності між вимогами застосунку і варіантами використання, яка, в свою чергу, показала що створений програмний продукт повністю виконує всі висунуті до нього вимоги. Були проаналізовані загальні вимоги, досліджена предметна область. Аналіз технічних рішень показав, що компанія Apple надає широкий спектр програмних продуктів, які можуть стати в нагоді для розв'язання задачі пов'язаної з OCR-системами. В результаті дослідження відомих програмних продуктів було знайдено невелику кількість програм, що відповідали б тематиці даного програмного забезпечення. В усіх проаналізованих додатках були описані як плюси так і мінуси.

					КПІ.ІП-6317.045490.02.81	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для створення якісного програмного продукту необхідно здійснити його попереднє моделювання та описати архітектуру. Для цього краще використовувати BPMN-діаграму (англ. Business Process Model and Notation) (Рисунок 2.1) [11]. Основною задачею BPMN -діаграм є моделювання бізнес-процесів, що є проміжним етапом між формалізацією/візуалізацією та втіленням бізнес процесу. Простіше кажучи, така нотація представляє собою опис графічних елементів, які використовуються для побудови схеми протікання бізнес-процесу.

Після запуску додатку, користувач має змогу вибрати одну з функцій:

- почати сканування нового тексту;
- перейти до налаштувань;
- перейти до мапи з позначеними відсканованими текстами;
- перейти до бібліотеки збережених текстів.

Далі в залежності від обраного шляху, користувач потрапляє або на екран сканування або до списку збережених текстів, мапи чи налаштувань де він може прослуховувати тексти або видаляти їх, відслідковувати на мапі.

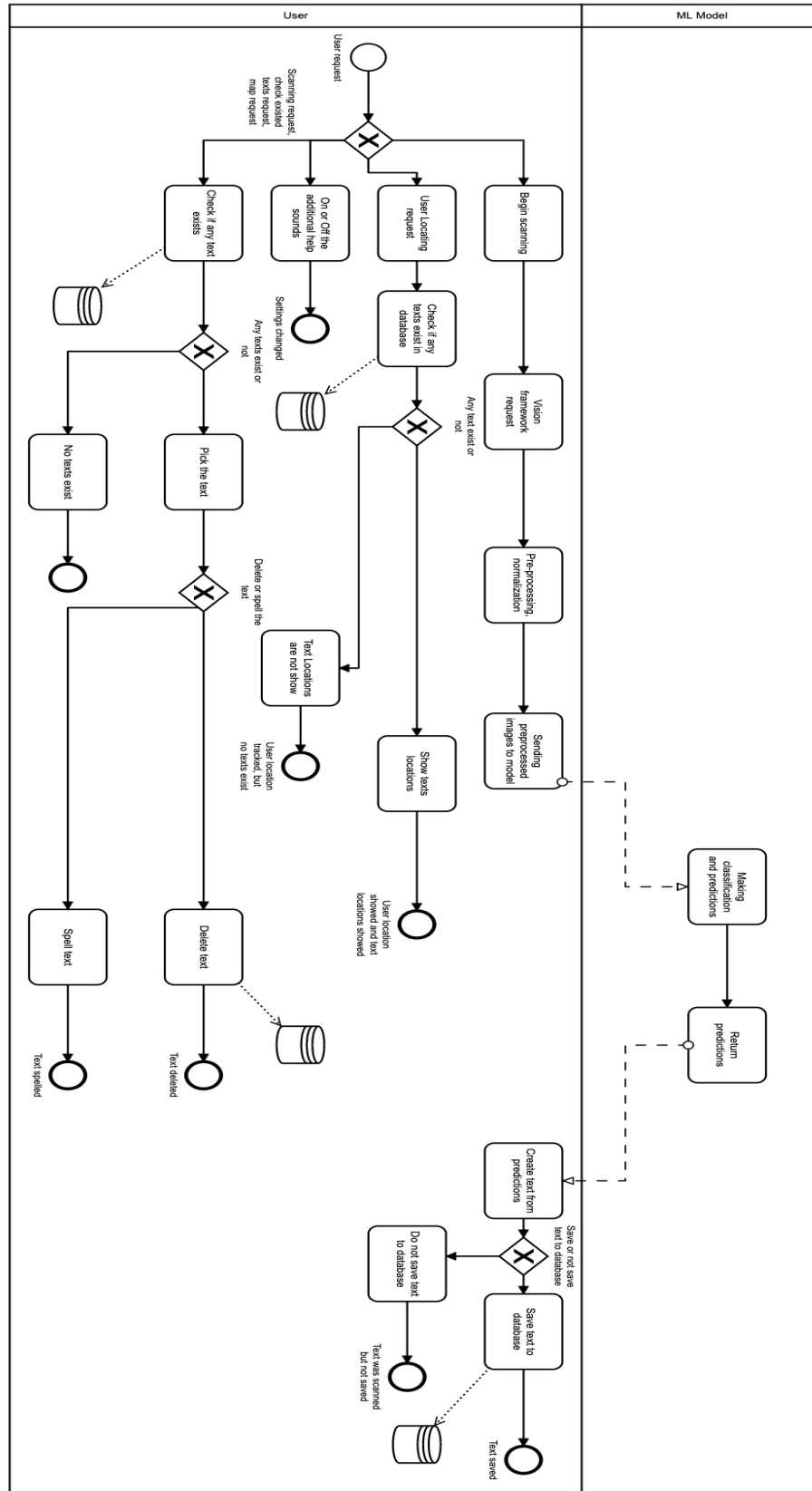


Рисунок 2.1 – Схема бізнес-процесів (BPMN)

2.2 Архітектура програмного забезпечення

Принцип взаємодії модулів зчитування та розпізнавання виконується в два етапи (Рисунок 2.2). Як можна побачити, сканований текст у вигляді зображень кожного окремого символу передається до тренованої моделі для розпізнавання

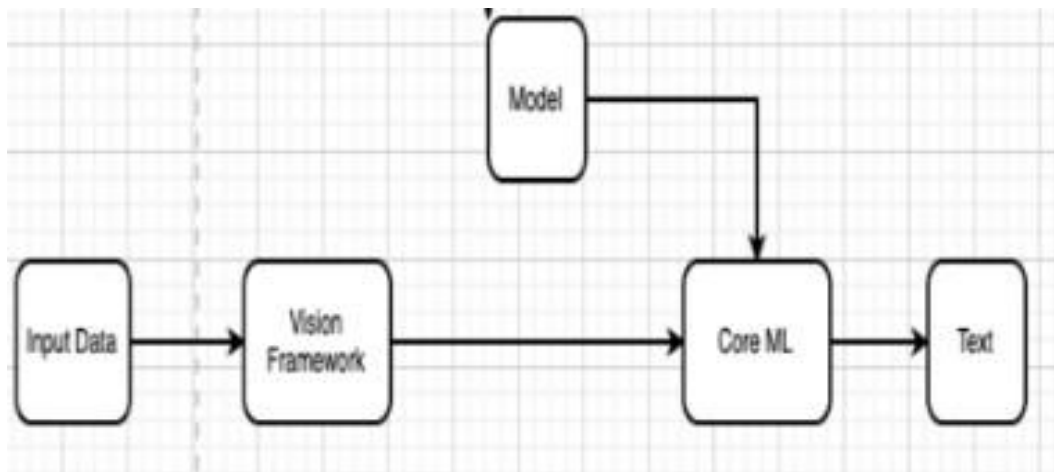


Рисунок 2.2 - Схема взаємодії Vision framework та Core ML

2.2.1 Зберігання даних

Тексти користувачів зберігаються за допомогою Core Data. Можна дивитись на цей фреймворк, як на оболочку над фізичним реляційним сховищем. Core Data використовує 4 типи сховищ які були описані раніше. Core Data має свій власний словник позначень:

- entity(сутність) – визначення класу в Core Data. В відношенні бази даних сутність відповідає таблиці;
- attribute(атрибут) – частина інформації, яка прикріплена до певної сутності. В базі даних сутність відповідає комірцю таблиці;
- relationship(зв'язок) – поєднання між декількома сутностями. В Core Data зв'язки які знаходяться між двома сутностями називаються to-one, а між декількома сутностями – to-many.

В даному застосуванні ми будемо використовувати лише In-memory формат (Рисунок 2.3).

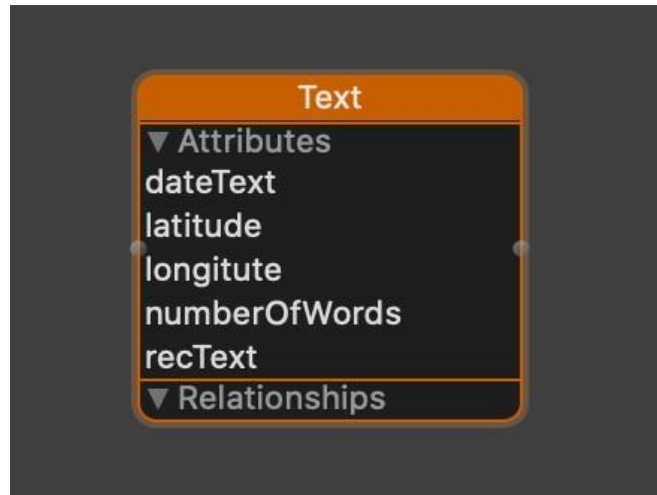


Рисунок 2.3 –Схема таблиці сканованих текстів

- latitude і longitude – атрибути типу Double, які вказують на координати де текст було прочитано;
- recText – атрибут типу String, відповідає за власний текст;
- dateText – атрибут типу Date, відповідає за дату, коли було розпізнано текст;
- numberOfWords – атрибут типу INT, вказує на кількість слів в даному тексті.

2.3 Конструювання програмного забезпечення

Наведено методи основних класів (Таблиця 2.1):

Таблиця 2.1 – Методи основних класів

Клас/Інтерфейс	Метод	Опис
ScanOrLibViewCon troller	onSettingsTapped()	Обробка натискання на кнопку Settings
	onMapTapped()	Обробка натискання на кнопку Map
	onScanTapped()	Обробка натискання на кнопку Scan
	onLibraryTapped()	Обробка натискання на кнопку Library

Продовження таблиці 2.1

	onInfoTapped()	Обробка натискання на кнопку Info
	soundInitiate()	Метод, що відповідає за звуковий супровід
ScanningViewController	scanDocument ()	Setup VNDocumentCameraViewController
	recognizeHandwritten()	Метод для розпізнавання handwritten символів
	processImage()	Оброблює зображення та виділяє його користувачу
	recognizeImage()	Метод, для розпізнавання символів отриманих після обробки
	onDeleteTapped ()	Видалення тексту

Продовження таблиці 2.1

	onSaveTapped ()	Зберігання тексту до бази
	Spell()	Метод, що відповідає за проговорення тексту
StartViewController	onTapOnScreen()	Відповідає за обробку тапу користувача по екрану та виконання подальшої дії.
	onInfoButtonTapped()	Відповідає за обробку натискання на кнопку Info.
	tapOnView()	Встановлення GestureRecognizer на backgroundImageView
LibraryCollectionView Controller	onInfoTapped()	Обробка натискання на infoButton
	configureCollectionView()	Встановлення конфігурацій collectionView
	configureCollectionViewCell	Встановлення конфігурацій collectionViewCell

Продовження таблиці 2.1

	onSaveTapped ()	Зберігання тексту до бази
	Spell()	Метод, що відповідає за проговорення тексту
	collectionViewDidScroll ()	Delegate метод класу UICollectionView
	collectionView(number OfItemsInSection)	DataSource метод класу UICollectionView
	collectionView(cellForItem)	DataSource метод класу UICollectionView
LibraryDetailsViewController	onDeleteTapped()	Видалення тексту
	onSpellTapped()	Проговорення тексту
	soundInitiate()	Додатковий звуковий супровід

Продовження таблиці 2.1

Text	fetchRequest():NSFetchRequest<Text>	Метод запиту на отримання даних CoreData
MapViewController	locate()	Визначення локації користувача
	makeAnnotation()	Встановлення анотацій на мапу
	setupList ()	Встановлення ListViewController до MapViewController
	locationManager(didUpdateLocation)	Реалізація методу CLLocationManagerDelegate
ListViewController	handleListTap()	Регулювання tap по List
	handleListPan()	Регулювання pan List
	updateInteractiveTransition()	Метод оновлення переходу, коли View перетягується

Продовження таблиці 2.1

AreaView	load(areas:[CGRect])	Встановлення обведень навколо тексту
	removeAreas()	Видалення обведень навколо тексту
	load(area: CGRect)	Відмальовування обведень навколо тексту
SoundModel	speakTheText(string: String)	Метод для проговорення тексту
SettingsViewController	onOnOffButtonTapped()	Метод для обробки дії при натисканні на кнопку On/Off
AdditionalSound Model	changeTheValue()	Метод, який змінює значення isAdditionalSound Enabled

Продовження таблиці 2.1

CoreDataManager	getData():Text	Метод, що відповідає за отримання даних CoreData
	writeData(text: Text)	Метод, що відповідає за зберігання даних CoreData
	deleteData(text: Text)	Метод, що відповідає за видалення даних CoreData

2.3 Архітектурний патерн

В якості архітектурного патерну було вирішено використовувати MVC-Model View Controller (Рисунок 2.5). Model – місце де зберігається вся бізнес логіка. Там знаходяться такі речі як парсери, менеджери, мережевий код і тд. View - представлення інтерфейсу користувача (UI). Controller – посередник між Model та View.

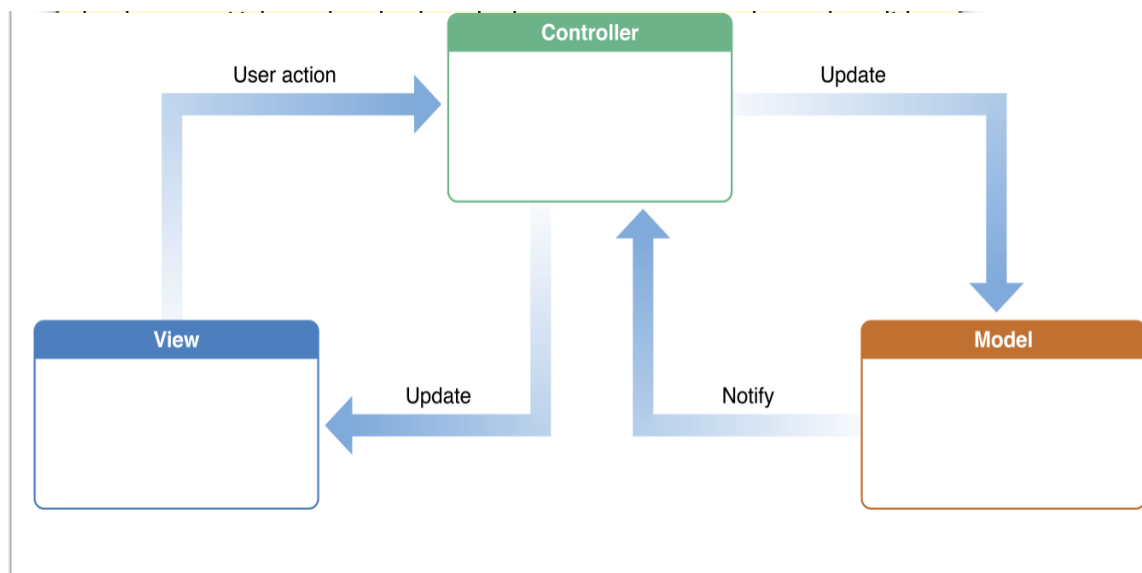


Рисунок 2.5 - Схема взаємодії Model-View-Controller

Працює така система досить просто: користувач взаємодіє із View що передається до Controller-у, де виконуються підрахунки. Controller міняє свій власний стан, що в свою чергу впливає на View. Якщо маніпуляції користувача вимагають зміни моделі, контролер викликає її для зміни. Зміни виконуються та проповідомляють контролері контролер знову змінює View. MVC було введено до застосування сама Apple. Хоча іноді, через MVC можуть виникати проблеми [9]. Це так звана проблема "Massive View Controller". Існує це через те, що MVC – погано масштабується. І використання його на великих проєктах не рекомендується.

2.4 Висновки до розділу

В даному розділі завдяки BPMN-діаграмі були змодельовані бізнес - процеси. Була представлена стратегія реалізації зберігання даних на пристрої. Описно архітектуру застосування. Було обгрунтовано використання фреймворку Core Data для зберігання сканованих та розпізнаних текстів .

Перелічені результати

трансформування моделі до формату mlmodel, в яких стало відомо, що точність моделі після виконання таких дій падає. Було запропоновано вирішення цієї проблеми шляхом застосування до отриманих слів функції AutoComplete. Разом з тим була представлена схема взаємодії Vision framework та Core ML. Було обрано архітектурний патерн – MVC. Названо його плюси та мінуси.

3 АНАЛІЗ ЯКОСТІ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Тестовий план

Підписка розробника IOS коштує \$ 99 на рік. Для створення і тестування програм в симуляторі вам не потрібно реєструватися. Тим не менш, ви повинні зареєструватися для тестування та розповсюдження програм на пристроях iOS. Реєстрація надає повний доступ до Центру розробки iOS і порталу налаштувань iOS. Тестовий план для визначення підходів, якими буде проведено тестування визначених об'єктів, створено дотримуючись стандарту IEEE 829 [10].

3.1.1 Вступ

Тестовий план описаний для мобільного застосування зі зчитування та розпізнання тексту для людей з вадами зору. Платформа додатку – IOS 13. Мета створення тестового плану полягає в наступному:

- визначити функції які будуть протестовані;
- визначити функції які не будуть протестовані;
- оприділити тести, що будуть виконуватися;
- визначити об'єкти тестування;
- оприділити критерії пройдених та не пройдених тестів;
- описати підхід до тестування.

3.1.2 Об'єкти тестування

Має бути протестовно мобільний додаток на платформі IOS. Тестування мобільного застосунку проводиться для версій IOS 13 та більш нових.

					КПІ.ІП-6317.045490.02.81	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.3 Функції що не обов'язково будуть тестуватися

Час за який виконується сканування тексту не має бути протестований. Не повинен також бути протестований час, за який виконується автонаведення на текст при скануванні. Швидкодія інтерфейсу користувача також може залишитись без тестування. Швидкодія роботи з базою даних не тестується.

3.1.4 Функції що обов'язково будуть тестуватися

Список функцій, які повинні протестуватися складається з таких:

- звуковий сигнал при успішному старті додатку;
- коректна робота кнопок виборудії;
- звуковий супровід при одному тапі на кнопку “Scanning”;
- звуковий супровід при одному тапі на кнопку “Library”;
- коректний перехід до екрану сканування при двох тапах на кнопку “Scanning”;
- коректний перехід до екрану списку збережених тестів при двох тапах на кнопку “Library”;
- звуковий супровід при одному тапі на кнопку “Spell”;
- коректне видалення тексту з бази при двох тапах на кнопку “Delete”;
- коректне зачитування тексту при двох тапах на кнопку “Spell”;
- коректне програвання звукового сигналу при наведенні на текст під час сканування;

– перевірка коректного збереження в базу даних тексту після сканування.

3.1.5 Підхід

Збір, аналіз тестових даних буде створюватися завдяки професійному інструменту Klaros Test Management. Тестувальник має стартувати тестина Klaros Test Management та помічати тести які з’явилися там як вдалі та невдалі. Також тестувальник має залишити відгук та пояснення щодо тестів. Людина, котра буде займатися тестуванням має створювати баг-репорти для невдало пройдених тестів.

3.1.6 Критерій пройдений/невдалий

Додаток не повинен мати критичних помилок, він має коректно виконувати методи затверджені в функціональних вимогах. Користувач повинен мати змогу без перешкод використовувати UI з підтримкою VoiceOver для всіх елементів користувацького інтерфейсу. Неприпустимі також помилки під час сканування тексту та його збереження до бази даних. Основні методи реалізовані для роботи з базою даних: збереження тексту, оновлення, видалення не повинні мати критичних зауважень. На момент закінчення тестування не повинно бути більше 7 нерозв’язаних проблем середнього масштабу. Мають бути реалізовані Unit-тестування функцій зазначені в функціональних вимогах.

3.1.7 Результат тестування

По завершенню тестування всі результати мають бути збережені в Klaros Test Management.

					КПІ.ІП-6317.045490.02.81	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.8 Задачі тестування

Обов'язкове виконання наступних пунктів:

- затверджено план тестування;
- описані функціональні вимоги;
- klaros Test Management налаштований на проведення тестування;
- додаток підготовлено до тестування;
- проходження тестів – успішне;
- підготовлено звіт з результатами за допомогою Klaros

TestManagement.

3.1.9 Вимоги до середовища

Додаток має бути запущений на реальному мобільному пристрої з версією IOS не нижче 13.

3.1.10 Відповідальність

Відповідальним за проведення тестування є розробник.

3.1.11 Затвердження

Рішення про завершення процесу тестування приймає розробник даного програмного забезпечення.

					КПІ.ІП-6317.045490.02.81	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Опис контрольного прикладу

Таблиця 3.1 - Звуковий сигнал при успішному старті додатку

Мета тесту	Коректна робота кнопок вибору дії
Схема виконання	Користувач натискає на “Sanning” та “Library”
Очікуваний результат	Користувач переходить на відповідні екрани
Дійсний результат	Користувач переходить на відповідні екрани

Таблиця 3.2 - Коректна робота кнопок вибору дії

Мета тесту	Коректна робота кнопок вибору дії
Схема виконання	Користувач натискає на “Sanning” та “Library”
Очікуваний результат	Користувач переходить на відповідні екрани
Дійсний результат	Користувач переходить на відповідні екрани

Таблиця 3.3- Звуковий супровід при одному тапі на кнопку “Scanning”

Мета тесту	Звуковий супровід при одному тапі на кнопку “Spell”;
Схема виконання	Користувач натискає на один раз “Spell ”
Очікуваний результат	Користувач чує синтезоване слово “Spell”
Дійсний результат	Користувач чує синтезоване слово “Spell”

Таблиця 3.4 - Звуковий супровід при одному тапі на кнопку “Delete”

Мета тесту	Звуковий супровід при одному тапі на кнопку “Delete”;
Схема виконання	Користувач натискає на один раз “Delete”
Очікуваний результат	Користувач чує синтезоване слово “Delete”
Дійсний результат	Користувач чує синтезоване слово “Delete”

Таблиця 3.5- Коректне зачитування тексту при двох тапах на кнопку “Spell”

Мета тесту	Коректне зачитування тексту при двох тапах на кнопку “Spell”
Схема виконання	Користувач натискає на два рази на “Spell”
Очікуваний результат	Користувач чує синтезований, правильний текст
Дійсний результат	Користувач чує синтезований, правильний текст

Таблиця 3.6- Коректне видалення тексту з бази при двох тапах на кнопку “Delete”

Мета тесту	Коректне видалення тексту з бази при двох тапах на кнопку “Delete”
Схема виконання	Користувач натискає два рази на “Delete”
Очікуваний результат	Текст видаляється з бази даних
Дійсний результат	Текст видаляється з бази даних

Таблиця 3.7- Коректне програвання звукового сигналу при наведенні на текст під час сканування

Мета тесту	Коректне програвання звукового сигналу при наведенні на текст під час сканування
Схема виконання	Користувач починає сканування і коли програма бачить текст програвается сигнал
Очікуваний результат	Сигнал програвается коректно
Дійсний результат	Сигнал програвается коректно

Таблиця 3.8- Перевірка коректного збереження в базу даних тексту

Мета тесту	Перевірка коректного збереження в базу даних тексту
Схема виконання	Користувач закінчує сканування
Очікуваний результат	Користувач бачить в “Library” збережений текст
Дійсний результат	Користувач бачить в “Library” збережений текст

3.2 Висновок до розділу

В даному розділі було створено тестовий план за стандартом IEEE829. Були описані такі моменти як: функції, що мають бути протестовані, план щодо тестування, задачі тестування, описано критерії пройдених/непроедених тестів. Було проведено тести, зазначення кобів'язков і до виконання. Результат показує, що тести успішно пройдені. Отже, даний мобільний застосунок є надійним.

					КПІ.ІП-6317.045490.02.81	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВПРОВАДЖЕННЯ ТА СУПРОВІДПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Через те, що розроблена програма є мобільним додатком, її розгортання з боку клієнта виглядає досить просто. Для початку, користувач повинен мати такі пристрої як смартфон “Iphone” з підтримкою ОС IOS не нижче ніж 13, або мобільний плеєр “Ipod touch” з підтримкою версії ОС IOS не нижче за 13. За умови, що перший пункт виконано, клієнт має 2 варіанти для завантаження представленого мобільного додатку – це напряму з магазину AppStore або бути запрошеним на бета-тестування через програму TestFlight. Але перший варіант в умовах дипломного проєкту не було здійснено, через те, що для вивантаження свого застосунку в даний магазин, розробник потребує річної підписки Apple Developer, яка коштує 99\$. Отже, завантаживши з AppStore безкоштовну програму TestFlight та отримавши запрошення на бета-тест, клієнт має можливість 89 днів користуватися та тестувати дипломний проєкт.

4.2 Робота з програмним забезпеченням

Керівництво користувача містить детальні інструкції по роботі з програмним забезпеченням.

4.3 Висновки до розділу

Даний розділ містить опис та перелік технічного обладнання, які мають бути задоволені користувачем, для успішного встановлення та використання розробленого програмного забезпечення. Також було створено керівництво користувача. Також, було детально описано процес розгортання створеного програмного забезпечення.

ВИСНОВКИ

В процесі написання даної дипломної роботи, було створено мобільне застосування зі зчитування тексту для людей з вадами зору. В роботі наявні 4 розділи.

Перший розділ містить опис та аналіз предметної області. Також, проводиться аналіз відомих технічних рішень, та ринок успішних аналогів. В даній частині зазначено функціональні та нефункціональні вимоги до програмного забезпечення.

В другому розділі було визначено з архітектурою програмного продукту, було описано конструювання додатку. Надано детальну схему класів та діаграму користувацького застосування.

В третьому розділі було розроблено детальний план тестування. Визначено об'єкти тестування, методи та задачі тестування. Представлено результати проведених тестів, за якими можна зробити висновок, що мобільний застосунок працює коректно та виконує все поставлені функціональні та нефункціональні задачі.

Четвертий розділ описує процес розгортання додатку з боку користувача. Також, створено BPMN схему застосування, проєктну документацію. Отже, можна зробити висновок, що зроблений програмний продукт може сильно допомогти в вирішенні питання отримання та сприймання текстової інформації людьми, що мають вади зору.

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Лісогор А.Ю. МОБІЛЬНЕ ЗАСТОСУВАННЯ ЗІ СЧИТУВАННЯ ТЕКСТУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ //IVВсеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020» Секція кафедри автоматизованих систем обробки інформації і управління. Матеріали конференції. – 2020. – С.
- 2) Оптичне розпізнавання символів [Електронний ресурс] // Wikipedia. – 2020. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Оптичне_розпізнавання_символів.
- 3) Serendi LTD. Pen to Print - Handwriting OCR [Електронний ресурс] / Serendi LTD // Apple Store. – 2020. – Режим доступу до ресурсу: <https://apps.apple.com/us/app/pen-to-print-handwriting-ocr/id1308003011>.
- 4) Coremltools [Електронний ресурс] // Coremltools. – 2017. – Режим доступу до ресурсу: <https://apple.github.io/coremltools/>.
- 5) Convolutional neural network [Електронний ресурс] // Wikipedia. – 2018. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Convolutional_neural_network.
- 6) VisionFramework [Електронний ресурс] // Apple Developer. – 2020. – Режим доступу до ресурсу: <https://developer.apple.com/documentation/vision>.
- 7) CoreML [Електронний ресурс] // Apple Developer. – 2020. – Режим доступу до ресурсу: <https://developer.apple.com/documentation/coreml>.
- 8) Keras [Електронний ресурс] // Keras.io. – 2018. – Режим доступу до ресурсу: <https://keras.io/about/>.
- 9) Model-View-Controller [Електронний ресурс] // Apple Developer. – 2006. – Режим доступу до ресурсу: <https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>.

КП.ІІІ-6317.045490.02.81

10) 829-2008 - IEEE Standard for Software and System Test Documentation [Електронний ресурс] // IEEE. – 2008. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/4578383>.

11) BPMN [Електронний ресурс] // Wikipedia. – 2020. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/BPMN>.

12) GovarthaniR. TextScanner (OCR) [Електронний ресурс] / RajeshGovarthani // AppleStore. – 2020. – Режим доступу до ресурсу: <https://apps.apple.com/us/app/text-scanner-ocr/id1225032527>.

13) ABBYY. FineScanner AI PRO-сканер OCR [Електронний ресурс] / ABBYY // Apple Store. – 2020. – Режим доступу до ресурсу: <https://apps.apple.com/ru/app/finescanner-ai-pro-сканер-ocr/id946930094>.

Змн.	Арк.	№ докум.	Підпис	Дата	63

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____Олександр ПАВЛОВ

“ ” _____2020р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ЗІ
ЗЧИТУВАННЯ ТЕКСТУ ДЛЯ ЛЮДЕЙ З
ВАДАМИ ЗОРУ

Опис програми

КП.ІП-6317.045490.03.13

“ПОГОДЖЕНО”

Керівник проєкту:

_____О.А.Халус

Нормоконтроль:

_____К.І.Ліщук

Виконавець:

_____А.Ю.Лісогор

Київ – 2020 року

ТЕКСТ ПРОГРАМНОГО КОДУ

Тексти програмного коду

Мобільне застосування зі зчитування тексту для людей з вадами зору

(Найменування програми (документа))

CD-R

(Вид носія даних)

14 арк, 11700 Кб

(Обсяг програми, арк., Кб)

Київ - 2020

					КПІ.ІП-6317.045490.03.13	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Extention Text

Text {

```
@nonobjc public class func fetchRequest() -> NSFetchRequest<Text> {
    return NSFetchRequest<Text>(entityName: "Text")
}
```

```
@NSManaged public var dateText: String?
```

```
@NSManaged public var latitude: Double
```

```
@NSManaged public var longitude: Double
```

```
@NSManaged public var numberOfWords: Int16
```

```
@NSManaged public var recText: String?
```

}

Class LibraryCollectionViewController

```
var itemsNumber = 1000
```

```
class LibraryCollectionViewController: UIViewController {
```

```
@IBOutlet weak var infoView: UIView!
```

```
@IBOutlet weak var collectionView: UICollectionView!
```

```
@IBOutlet weak var selectTextButton: UIButton!
```

```
@IBOutlet weak var pageControl: UIPageControl!
```

```
@IBAction func onInfoTapped(_ sender: Any) {
```

```
    animationOne()
```

```
}
```

					КПІ.ІП-6317.045490.03.13	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

let collectionViewCellHeightCoefficient: CGFloat = 0.85
    let collectionViewCellWidthCoefficient: CGFloat = 0.55
    let priceButtonCornerRadius: CGFloat = 10
    let gradientFirstColor = UIColor(hex: "ff8181").cgColor
    let gradientSecondColor = UIColor(hex: "a81382").cgColor
    let cellsShadowColor = UIColor(hex: "2a002a").cgColor
    let productCellIdentifier ="ProductCollectionViewCell"

    override func viewDidLoad() {
        super.viewDidLoad()

        configureCollectionView()
        infoView.layer.cornerRadius = 5
    }

    private func configureCollectionView() {
        let gravitySliderLayout = GravitySliderFlowLayout(with:
CGSize(width: collectionView.frame.size.height *
collectionViewCellWidthCoefficient, height: collectionView.frame.size.height
* collectionViewCellHeightCoefficient))
        collectionView.collectionViewLayout = gravitySliderLayout
        collectionView.dataSource = self
        collectionView.delegate = self
    }

    private func configureProductCell(_ cell: UICollectionViewCell, for
indexPath: IndexPath) {
        cell.clipsToBounds = false
        cell.layer.cornerRadius = 7
        let gradientLayer = CAGradientLayer()
        gradientLayer.frame = cell.bounds
        gradientLayer.colors = [gradientFirstColor, gradientSecondColor]
        gradientLayer.cornerRadius = 21
        gradientLayer.masksToBounds = true
        cell.layer.insertSublayer(gradientLayer, at: 0)

        cell.layer.shadowColor = cellsShadowColor
        cell.layer.shadowOpacity = 0.2
        cell.layer.shadowRadius = 20
        cell.layer.shadowOffset = CGSize(width: 0.0, height: 30)

        cell.textField.text = texts[indexPath.row % texts.count]

```

					КП.ІП-6317.045490.03.13	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        cell.titleLabel.layer.cornerRadius = 8
        cell.titleLabel.clipsToBounds = true
        cell.titleLabel.text = titles[indexPath.row % titles.count]
        cell.titleLabel.layer.borderColor = UIColor.white.cgColor
        cell.titleLabel.layer.borderWidth = 1.0
    }

    private func animateChangingTitle(for indexPath: IndexPath) {
        UIView.transition(with: productTitleLabel, duration: 0.3, options:
        .transitionCrossDissolve, animations: {
            self.productTitleLabel.text = self.titles[indexPath.row %
self.titles.count]
        }, completion: nil)
        UIView.transition(with: productSubtitleLabel, duration: 0.3,
options: .transitionCrossDissolve, animations: {
            self.productSubtitleLabel.text = self.subtitles[indexPath.row %
self.subtitles.count]
        }, completion: nil)
        UIView.transition(with: priceButton, duration: 0.3, options:
        .transitionCrossDissolve, animations: {
            self.priceButton.setTitle(self.prices[indexPath.row %
self.prices.count], for: .normal)
        }, completion: nil)
    }

    private func animationOne()
    {
        let
        animationElement1 =
        CAAAnimationGroup()

        animationElement1.isRemoved
        OnCompletion = false

        animationElement1.fillMode
        = .removed

        animationElement1.duration
        =5.65

        animationElement1.animation
        s =[]
    }

```

					КПІ.ІП-6317.045490.03.13	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        let
        animationProperty1 =
        CAKeyframeAnimation()

        animationProperty1.beginTime = 0

        animationProperty1.repeatCount = Float(1)

        animationProperty1.duration = 1

        animationProperty1.autoreverses = false

        animationProperty1.isRemovedOnCompletion = false

        animationProperty1.timingFunction =
        CAMediaTimingFunction(controlPoints: 0.22, 0.61, 0.61, 1)

        animationProperty1.keyPath = "transform.translation.x"

        animationProperty1.keyTimes = [ 0, 0.6, 0.75, 0.9, 1 ]

        animationProperty1.values = [ -3000, 25, -10, 5, 0 ]

        animationElement1.animations?.append(animationProperty1)

```

```

        let
        animationProperty2 =
        CAKeyframeAnimation()

        animationProperty2.beginTime = 0

        animationProperty2.repeatCount = Float(1)

        animationProperty2.duration = 5.65

        animation

```

					КПІ.ІП-6317.045490.03.13	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Property2.autoreverses =
false

animationProperty2.isRemovedOnCompletion = false

animationProperty2.timingFunction =
CAMediaTimingFunction(controlPoints: 0.22, 0.61, 0.61, 1)

animationProperty2.keyPath = "opacity"

animationProperty2.keyTimes = [ 0, 0.11, 1 ]

animationProperty2.values = [ 0, 0.35, 0.35 ]

animationElement1.animations?.append(animationProperty2)

self.infoView.layer.add(animationElement1, forKey: nil)
}

}

extension
LibraryCollectionViewController:
UICollectionViewDataSource
{
    func collectionView(_ collectionView:
UICollectionView,
numberOfItemsInSection section: Int) -> Int {
        return itemsNumber
    }

    func collectionView(_ collectionView:
UICollectionView,
cellForItemAt indexPath: IndexPath) ->

```

					КПІ.ІП-6317.045490.03.13	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		


```

UICollectionViewCell {
    let cell =
collectionView.dequeueReusableCell(withReuseIdentifier
: "MyCollectionViewCell",
for: indexPath) as!
UICollectionViewCell

self.configureProductCell(cell, for: indexPath)
    return cell
}

}

extension
LibraryCollectionViewContro
ller:
UICollectionViewDelegate {
    func
scrollViewDidScroll(_
scrollView: UIScrollView) {
        let locationFirst =
CGPoint(x:
collectionView.center.x +
scrollView.contentOffset.x,
y:collectionView.center.y
+
scrollView.contentOffset.y)
        letlocationSecond
= CGPoint(x:
collectionView.center.x +
scrollView.contentOffset.x
+ 10, y:
collectionView.center.y +
scrollView.contentOffset.y)
        let locationThird =
CGPoint(x:
collectionView.center.x +
scrollView.contentOffset.x
- 10, y:
collectionView.center.y +
scrollView.contentOffset.y)

        if let
indexPathFirst =
collectionView.indexPathFor
Item(at: locationFirst),
let indexPathSecond =
collectionView.indexPathFor
Item(at: locationSecond),
let indexPathThird =
collectionView.indexPathFor
Item(at: locationThird),
indexPathFirst.row ==
indexPathSecond.row &&
indexPathSecond.row ==
indexPathThird.row &&
indexPathFirst.row !=
pageControl.currentPage{

```

					КП.ІП-6317.045490.03.13	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

```

StartViewController

class StartViewController: UIViewController {

    @IBOutlet weak var imageViewBackgroundScreen: UIImageView!
    override func viewDidLoad() {
        super.viewDidLoad()
        tapOnView()
        tapInfoView.layer.cornerRadius = 5
        // Do any additional setup after loading the view.
    }

    @IBOutlet weak var tapInfoView: UIView!
    @IBAction func onInfoButtonTapped(_ sender: Any) {
        animationOne()
    }

    func tapOnView() {
        let imageGesture = UITapGestureRecognizer(target: self,
action: #selector(onTapOnScreen))

self.imageViewBackgroundScreen.addGestureRecognizer(imageGesture)

    }

    @objc func onTapOnScreen() {
        let storyboard = UIStoryboard(name: "Main", bundle: nil)
        let secondVC =
storyboard.instantiateViewController(identifier: "ScanOrLibViewController")
        secondVC.modalPresentationStyle = .fullScreen
        present(secondVC, animated: true, completion: nil)
    }

    privatefunc animationOne() {
        let animationElement1 = CAAAnimationGroup()
        animationElement1.isRemovedOnCompletion = false
        animationElement1.fillMode = .removed
        animationElement1.duration = 5.65
        animationElement1.animations = []

        let animationProperty1 = CAKeyframeAnimation()
        animationProperty1.beginTime = 0
        animationProperty1.repeatCount = Float(1)
        animationProperty1.duration = 1
        animationProperty1.autoreverses = false
        animationProperty1.isRemovedOnCompletion = false
        animationProperty1.timingFunction =
CAMediaTimingFunction(controlPoints: 0.22, 0.61, 0.61, 1)
        animationProperty1.keyPath = "transform.translation.x"
        animationProperty1.keyTimes = [ 0, 0.6, 0.75, 0.9, 1 ]
        animationProperty1.values = [ -3000, 25, -10, 5, 0 ]
        animationElement1.animations?.append(animationProperty1)

        let animationProperty2 = CAKeyframeAnimation()

```

```

        animationProperty2.beginTime = 0
        animationProperty2.repeatCount = Float(1)
        animationProperty2.duration = 5.65
        animationProperty2.autoreverses = false
        animationProperty2.isRemovedOnCompletion = false
        animationProperty2.timingFunction =
CAMediaTimingFunction(controlPoints: 0.22, 0.61, 0.61, 1)
        animationProperty2.keyPath = "opacity"
        animationProperty2.keyTimes = [ 0, 0.11, 1 ]
        animationProperty2.values = [ 0, 0.35, 0.35 ]
        animationElement1.animations?.append(animationProperty2)

        self.tapInfoView.layer.add(animationElement1, forKey: nil)
    }

}

```

```

VNDocumentCameraViewControllerDelegate

func documentCameraViewController(_
controller: VNDocumentCameraViewController,
didFinishWith scan: VNDocumentCameraScan)
{Make sure the user scanned at least one
page

    guard scan.pageCount >= 1 else {

        You are responsible for dismissing the
VNDocumentCameraViewController.

        controller.dismiss(animated:
true)

        return
    }
}

```

This is a workaround for the VisionKit bug which breaks the `UIImage` returned from `VisionKit`

See the `Image Loading Hack` section below for more information.

```
let originalImage =
scan.imageOfPage(at: 0)

let fixedImage =
reloadedImage(originalImage)
```

You are responsible for dismissing the VNDocumentCameraViewController.

```
controller.dismiss(animated:
true)
```

```
playSoundScan()
```

```
Process the image
processImage(fixedImage)
}
```

```
func documentCameraViewController(_
controller: VNDocumentCameraViewController,
didFailWithError error: Error) {
```

The VNDocumentCameraViewController failed with an error.

For now, we'll print it, but you should handle it appropriately in your app

					КПІ.ІП-6317.045490.03.13	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        print(error)

        // You are responsible for
        dismissing the
        VNDocumentCameraViewController.

        controller.dismiss(animated:
true)
    }

    func
    documentCameraViewControllerDidCancel(_
controller: VNDocumentCameraViewController)
    {
        You are responsible for
        dismissing the
        VNDocumentCameraViewController.

        controller.dismiss(animated:
true)
    }

```

MARK: - Image Loading Hack

VisionKit currently has a bug where the images returned reference unique files on disk which are deleted after dismissing the VNDocumentCameraViewController.

To work around this, we have to create a new UIImage from the data of the original image from VisionKit.

I have filed a bug (FB6156927) -

					КПІ.ІП-6317.045490.03.13	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    hopefully this is fixed soon.

    @IBAction func onDeleteTapped(_
sender: Any) {
        }

    @IBAction func onSaveTapped(_
sender: Any) {
        }

    func reloadedImage(_ originalImage:
UIImage) -> UIImage {
        guard let imageData =
originalImage.jpegData(compressionQuality:
1),
            let reloadedImage =
UIImage(data: imageData) else {
            return originalImage
        }
        return reloadedImage
    }

}

// MARK: - Delete and save to CoreData

func playSoundScan() {
    guard let url =
Bundle.main.url(forResource: "Recognition",
withExtension: "mp3") else { return }

    do {
        try
AVAudioSession.sharedInstance().setCategory

```

					КПІ.ІП-6317.045490.03.13	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

```
(.playback, mode: .default)

    try

AVAudioSession.sharedInstance().setActive(tr
ue)

    playertwo = try

AVAudioPlayer(contentsOf: url, fileTypeHint:
AVFileType.mp3.rawValue)

    player = try

AVAudioPlayer(contentsOf: url, fileTypeHint:
AVFileTypeMPEGLayer3)

    guard let player = playertwo
else { return }

    player.play()

    } catch let error {

print(error.localizedDescription)

    }

}
```

					КПІ.ІП-6317.045490.03.13	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____Олександр ПАВЛОВ

“ ” _____ 2020р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ЗІЗЧИТУВАННЯ
ТЕКСТУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ

Технічне завдання

КП.ІП-6317.045490.04.91

“ПОГОДЖЕНО”

Керівник проєкту:

_____О.А.Халус

Нормоконтроль:

_____К.І.Ліщук

Виконавець:

_____А.Ю.Лісогор

Київ – 2020 року

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
4.1	Вимоги до функціональних характеристик	6
4.2	Вимоги до надійності	6
4.3	Умови експлуатації	6
4.4	Вимоги до складу і параметрів технічних засобів	7
4.5	Вимоги до інформаційної та програмної сумісності	7
4.6	Вимоги до маркування та пакування	7
4.7	Вимоги до транспортування та зберігання... ..	7
4.8	Спеціальні вимоги... ..	7
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....	8
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	9
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	10

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Мобільне застосування зі зчитування тексту для людей з вадами зору

Галузь застосування: Повсякденне чи професійне життя.

Мобільне застосування зі зчитування тексту для людей з вадами зору, надає їм можливість сприймати інформацію з текстових істочників. Додаток створено з урахуванням потреб такої категорії людей. Виконує зчитування, аналіз, зберігання та зачитування тексту.

					КПІ.ІП-6317.045490.04.91	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПІДСТАВА ДІЯРОЗРОБКИ

Підставою дія розробки мобільного застосування зчитування тексту для людей з вадами зору є завдання на дипломне проєктування, затверджене кафедрою автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

					КПІ.ІП-6317.045490.04.91	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для надання можливості людям з вадами зору сприймати текстову інформацію, що значно полегшує повсякденне життя такої людини. Використання шрифту Брайля не покриває весь комплекс задач із передачі текстової інформації людям з проблемами зору. Даний мобільний додаток має допомогти в цьому.

Метою розробки є надання функціоналу людям з проблемами зору, такого що дозволить їм не лише зчитувати текстову інформацію, а й зберігати та зачитувати отримані після сканування результати.

					КПІ.ІП-6317.045490.04.91	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1.1 Для користувача:

- зчитування та розпізнавання тексту в режимі камери;
- автовизначення кордону тексту;
- підтримка UI функцій VoiceOver;
- збереження тексту;
- зачитування тексту;
- видалення текстів.

4.1.2 Розробку виконати на платформі macOS.

4.1.3 Додаткові вимоги

Простий та зрозумілий користувацький інтерфейс.

4.2 Вимоги до надійності

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.2.3 Забезпечити цілісність інформації в базі даних.

					КПІ.ІП-6317.045490.04.91	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96. Невисуваються.

4.3.2 Обслуговування

Додаткове обслуговування не передбачено.

4.3.3 Обслуговуючий персонал Потреба не зазначається

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно функціонувати на пристроях, які керуються операційною системою IOS не нижче 13 версії.

4.4.2 Мінімальна конфігурація технічних засобів:

4.4.2.1 Тип процесору Apple A7.

4.4.2.2 Об'єм ОЗП 1 Гб.

4.4.2.3 Об'єм внутрішньої пам'яті – 1.5 Гб.

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційної системи IOS 13 чи більш актуальних.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі: стрімінг відео з відеокамери мобільного пристрою.

4.5.3 Результати повинні бути представлені в наступному форматі: текстова інформація.

4.5.4 Програмне забезпечення повинно оброблювати інформацію під час стрімінгу відео та зберігати вихідні дані у вигляді тексту в пам'яті пристрою.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Тестування можливе лише за наявності підписки Apple-розробника та при наявності мобільного пристрою з операційною системою IOS версії 13 чи вище.

					КПІ.ІП-6317.045490.04.91	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Розроблювані програмні модулі повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 60 аркушах формату А4 (без додатків 5.3.2 -5.3.4).

5.3.2 Технічне завдання.

5.3.3 Керівництво користувача.

5.3.4 Програма та методика тестування

5.4 Графічна частина повинна бути виконана на аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки

5.4.1 Схема структурна варіантів використання.

5.4.2 Схема структурна класів програмного забезпечення

5.4.3 Креслення вигляду екранних форм.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення рекомендованої літератури	19.03.2020	
2.	Аналіз існуючих методів розв'язання задачі	26.03.2020	
3.	Постановка та формалізація задачі	26.03.2020	Технічне завдання
4.	Аналіз вимог до програмного забезпечення	02.04.2020	Схема структура варіантів використання, вимоги до програмного забезпечення
5.	Алгоритмізація задачі	02.04.2020	
6.	Моделювання програмного забезпечення	09.04.2020	
7.	Обґрунтування використовуваних технічних засобів	16.04.2020	
8.	Розробка архітектури програмного забезпечення	23.04.2020	Схема структурна класів програмного забезпечення
9.	Розробка програмного забезпечення	30.04.2020	Тексти програмного забезпечення

10.	Налагодження програми	07.05.2020	Програма та методика тестування
11.	Виконання графічних документів	14.05.2020	Графічний матеріал проекту
12.	Оформлення пояснювальної записки	21.05.2020	Пояснювальна записка проекту
13.	Подання ДП на попередній захист	28.05.2020	
14.	Подання ДП рецензенту	03.05.2020	
15.	Подання ДП на основний захист	08.06.2020	

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-6317.045490.04.91	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ **Олександр ПАВЛОВ**

“ ” _____ 2020р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ЗІ
ЗЧИТУВАННЯ ТЕКСТУ ДЛЯ ЛЮДЕЙ З
ВАДАМИ ЗОРУ

Програма та методика тестування

КПІ.ІП-6317.045490.05.51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ **О.А.Халус**

Нормоконтроль:

_____ **К.І.Ліщук**

Виконавець:

_____ **А.Ю.Лісогор**

ЗМІСТ

1	ОБ’ЄКТИПРОБУВАНЬ.....	3
2	МЕТА ТЕСТУВАННЯ.....	4
3	МЕТОДИ ТЕСТУВАННЯ.....	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ.....	6

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Мобільне застосування зі зчитування тексту для людей з вадами зору. Програмний продукт, що створений мовою Swift із застосованим архітектурним патерном MVC. Додатком підтримується версія IOS - не менше 13.

					КПІ.ІП-6317.045490.05.51	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

2 МЕТА ТЕСТУВАННЯ

Під час тестування тестуються наступні функції:

- звуковий супровід основних етапів використання додатку;
- коректне виконання функції зчитування тексту;
- коректне виконання функції розпізнавання тексту;
- функція видалення тексту;
- функція зберігання тексту;
- коректне функціонування бібліотеки текстів;
- коректне виконання зачитування текстів;
- коректне функціонування карти з позначеними на ній місцями, де тексти були скановані;
- тестування UI.

					КПІ.ІП-6317.045490.05.51	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

3 МЕТОДИ ТЕСТУВАННЯ

Тестування проводиться методом напівпрозорого ящика, що в свою чергу, має на увазі тестування внутрішньої структури, основних функцій та алгоритмів, але з точки зору користувача. Тестувальник вивчає код програми для того, щоб створити відповідний тест-кейс для подальшої перевірки функціональності додатку. Наступні види тестування використані:

- юзабіліті тестування для перевірки на відповідність вимогам зручності використання застосунку;
- функціональне тестування для перевірки на відповідність вимогам до функціоналу застосунку;
- конфігураційне тестування для перевірки того, як буде поводити себе програмний продукт при значній зміні конфігурації;
- тестування UI.

					КПІ.ІП-6317.045490.05.51	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Юзабіліті тестування проводиться емпіричним методом. Конфігураційне тестування та тестування UI виконуються методом кроссапаратного тестування шляхом зміни пристроїв на яких завантажено даних додаток . Функціональне тестування впроваджено шляхом модульного тестування, тобто тестування кожної атомарної функціональності застосунку окремо в штучно створеному середовищі.

					КПІ.ІП-6317.045490.05.51	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____Олександр ПАВЛОВ

“ ” _____2020р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ЗІ ЗЧИТУВАННЯ
ТЕКСТУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ

Керівництво користувача

КП.ІП-6317.045490.06.34

“ПОГОДЖЕНО”

Керівник проєкту:

_____О.А.Халус

Нормоконтроль:

_____К.І.Ліщук

Виконавець:

_____А.Ю.Лісогор

Київ – 2020 року

Після успішного завантаження додатку з магазину AppStore, лого даного додатку з'явиться в меню користувача. Використання створеного застосунку можна умовно поділити на 2 типи:

- з ввімкненою функцією VoiceOver;
- з вимкненою функцією VoiceOver.

Якщо VoiceOver в девайсі користувача вимкнено, то для запуску необхідно лише натиснути на логотип в меню. Після запуску додатку, юзер потрапляє на екран привітання (Рисунок 1) та після тапу по екрану на меню вибору опції (Рисунок 1). В перші декілька секунд після запуску, можна почути синтезоване привітання. На даному екрані наявні 5 варіантів дій:

- перехід до бібліотеки збережених текстів шляхом натискання на кнопку “Library”;
- перехід до екрану сканування шляхом натискання на кнопку “Scan”;
- перехід до екрану з мапою шляхом натискання на кнопку “Map”;
- перехід до налаштувань шляхом натискання на круглу кнопку в нижній частині екрану;
- отримання додаткової інформації шляхом натискання на кнопку з зображенням знака оклику.

В випадку, коли користувач натиснув на кнопку Library (Рисунок 2), він попадає до бібліотеки. За наявності збережених текстів, на даному екрані наявний scrollable елемент, що представляє собою колонки з вирізками текстів, та основною інформацією про них. Користувач має проскролити та вибрати текст, після чого, натиснувши на кнопку з позначкою олівця переходить на екран з детальною інформацією про проданий текст. На даному екрані також наявні звукові позначення та кнопка із знаком оклику, натиснувши на яку, можна отримати додаткову інформацію. Повернутися до меню вибору опцій можна смахнувши зверху вниз.

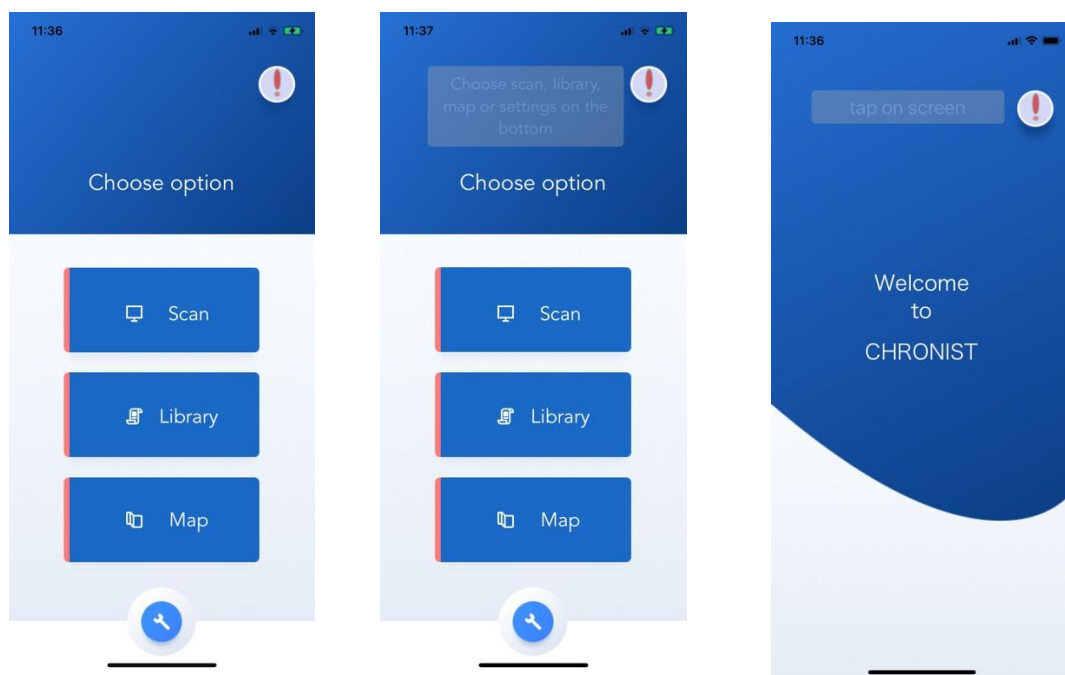


Рисунок 1 – Стартовий екран та екран вибору опції

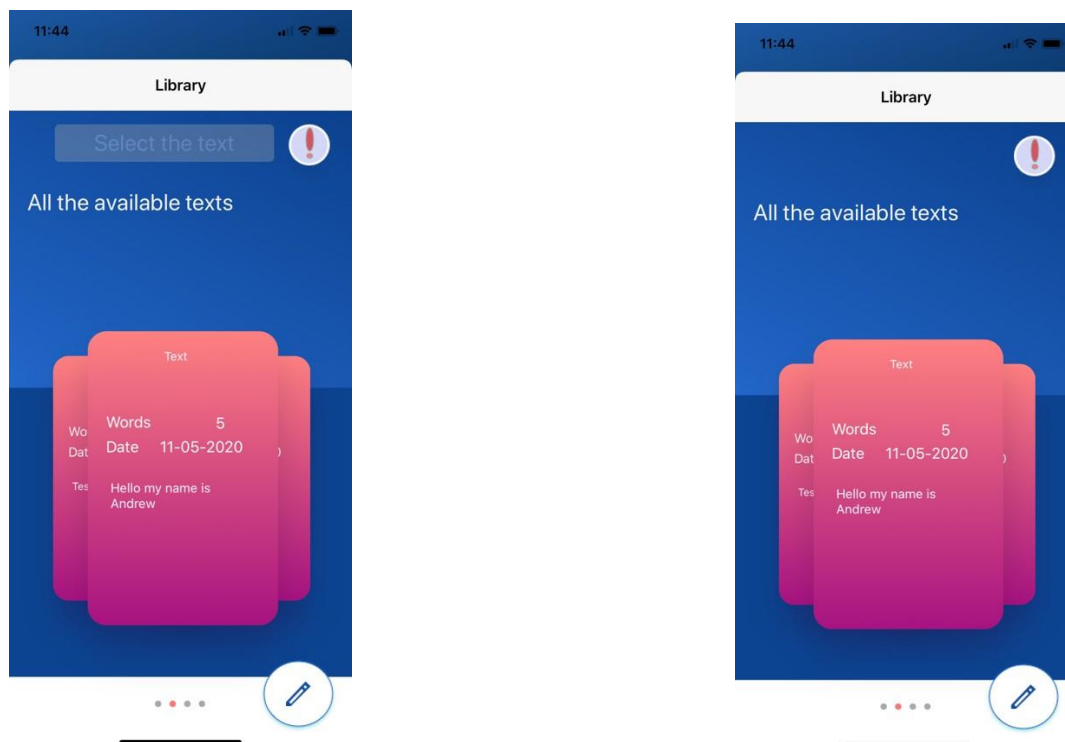


Рисунок 2 –Екран бібліотеки

Після вибору тексту та натискання на кнопку із позначкою олівця, користувач потрапляє на екран з детальною інформацією про обраний текст (Рисунок 3). Тут представлені:

- власнотекст;
- інформація про кількістьслів;
- інформація про дату збереженнятексту;
- зачитування тексту при натисканні на кнопку згалочкою;
- видаленнятекступринатисканніначервонукнопкузпозначкоюсмітник

а. Повернутися на попередній екран можна потягнувши актуальний зверху вниз.

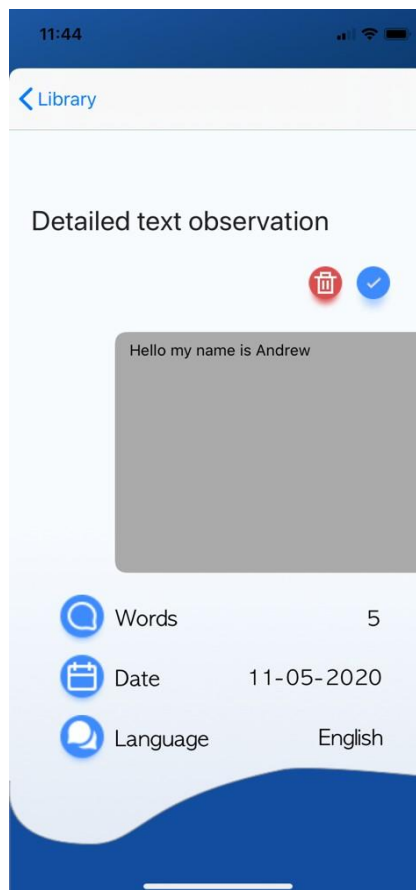


Рисунок 3 –Екран детальної інформації про текст

					КПІ.ІП-6317.045490.06.34	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо в меню вибору опції користувачем було натиснуто на кнопку Scan, відповідно виконується перехід на екран сканування. На даному екрані представлено:

- зображення з виділеним текстом;
- власне розпізнаний текст;
- кнопка збереження тексту до бібліотеки;
- кнопка зачитування тексту;
- кнопка сканування.

Після натискання на кнопку сканування виконується перехід на екран камери, де виконується власне сканування (Рисунок 5). Для виконання даного процесу

користувачу необхідно навести камеру на текст. Коли камера бачить текст, юзер чує звукові сигнали. Текст фотографується автоматично. Тепер необхідно натиснути на кнопку save в правому нижньому куті (Рисунок 4). І автоматично виконується зворотній перехід до екрану сканування, де користувач бачить розпізнаний текст, може його прослухати, зберегти. Для виходу з даного режиму потрібно змахнути актуальний екран зверху-вниз.

Якщо в меню вибору опції користувачем було натиснуто на кнопку Map, відповідно виконується перехід на екран з мапою. На даному екрані представлено:

- мапу;
- таблицю з координатами збережених текстів.

На даному екрані користувач має можливість побачити на мапі координати всіх збережених текстів.

Таблиця з додатковою інформації відкривається шляхом перетягування її з нижньої частини екрану (Рисунок 6). Вихід до попереднього - змахнути актуальний екран зверху-вниз.



Рисунок 4 –Екран сканування тексту

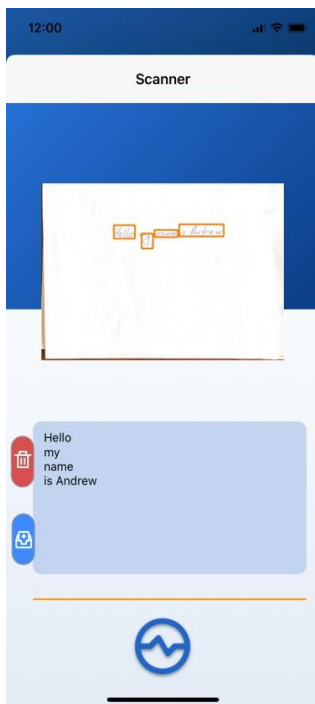


Рисунок 5 –Екран відсканованого тексту

					КПІ.ІП-6317.045490.06.34	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

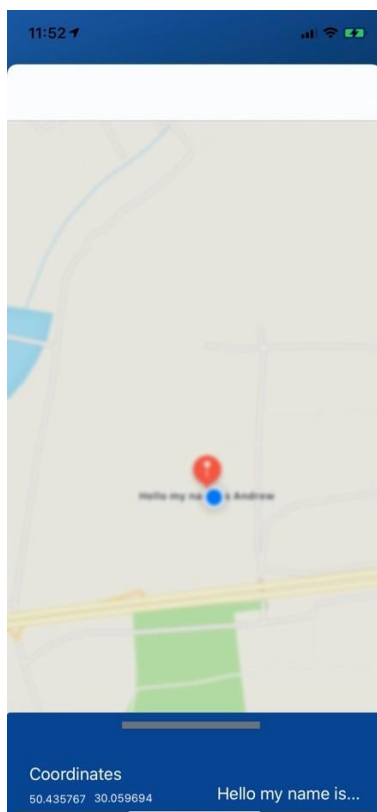
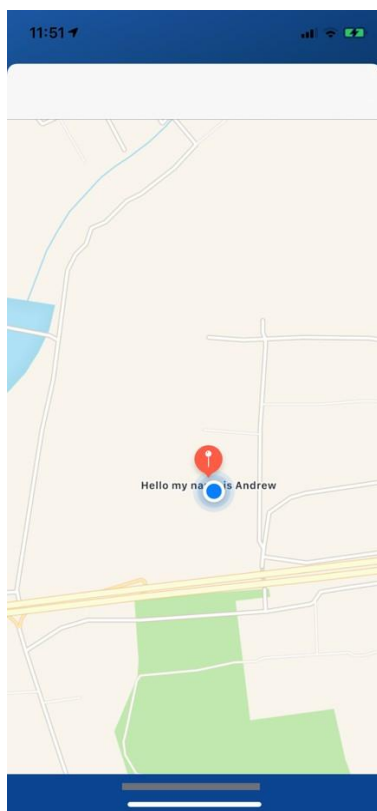


Рисунок 6 –Екран мапи

					КПІ.ІП-6317.045490.06.34	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Остання опція на екрані меню – налаштування. Натиснувши круглу кнопку в нижній частині екрану – користувач переходить в меню налаштувань. Наданий момент налаштування містять лише одну кнопку з On/Off додаткових звукових позначень (Рисунок 7). Повернутися до основного меню можна махнувши актуальний екран зверху-вниз.

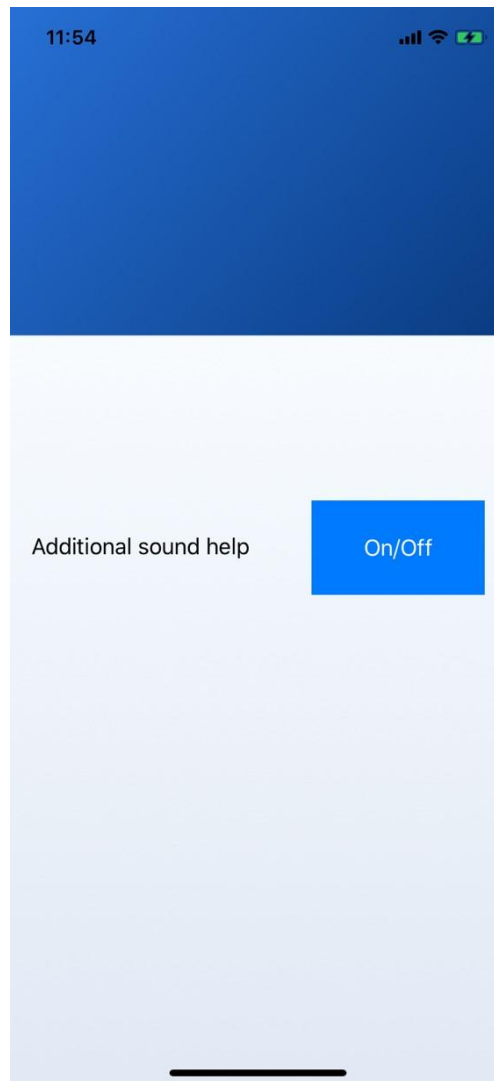


Рисунок 7 – Екран налаштувань

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____Олександр ПАВЛОВ

“ ” _____2020р.

МОБІЛЬНЕ ЗАСТОСУВАННЯ ЗІ ЗЧИТУВАННЯ
ТЕКСТУ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ

Графічний матеріал
КП.ІП-6317.045490.07.99

“ПОГОДЖЕНО”

Керівник проєкту:

_____О.А.Халус

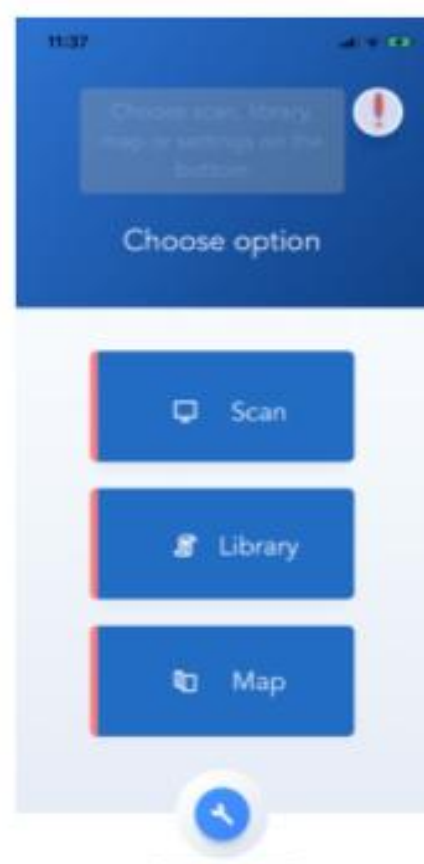
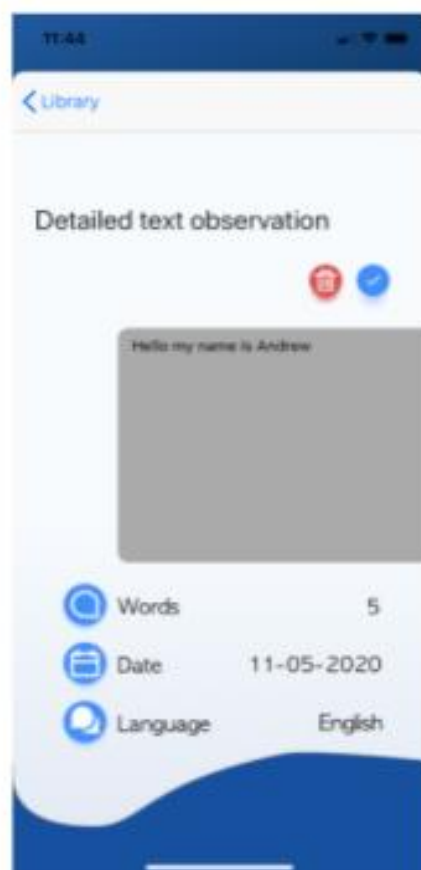
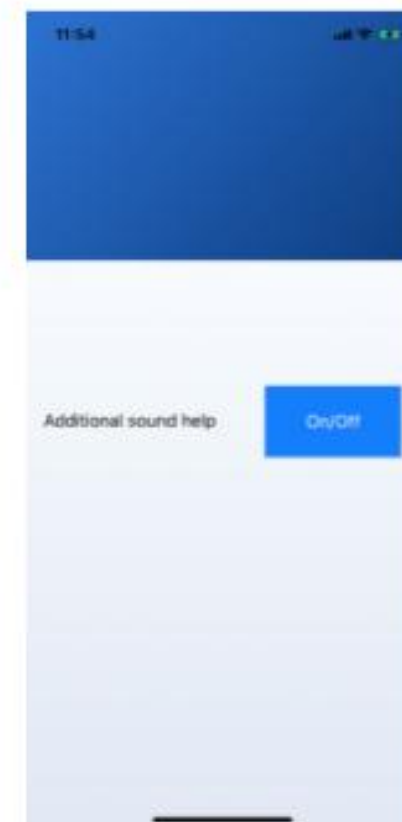
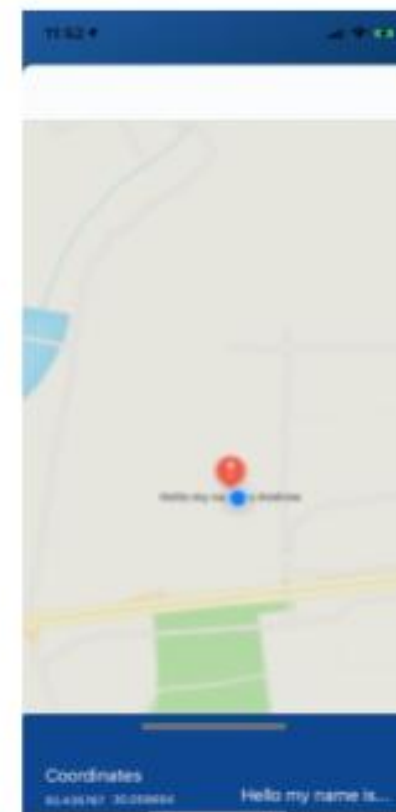
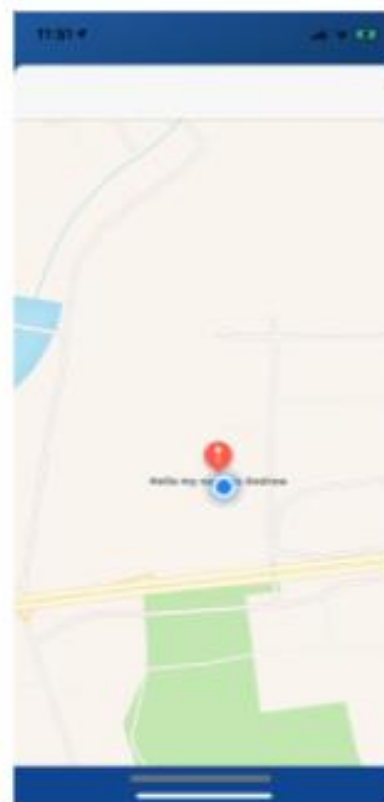
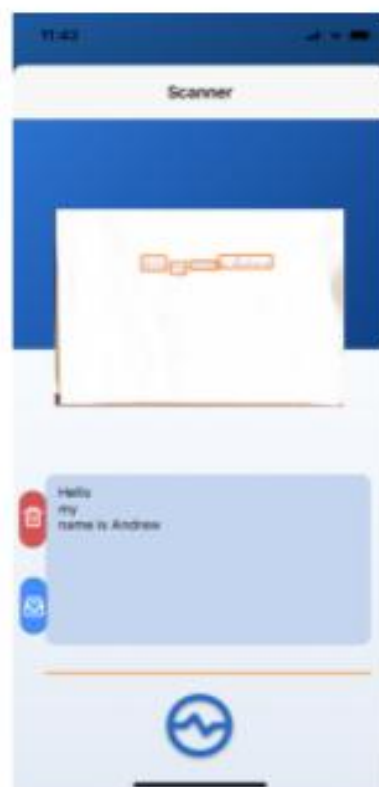
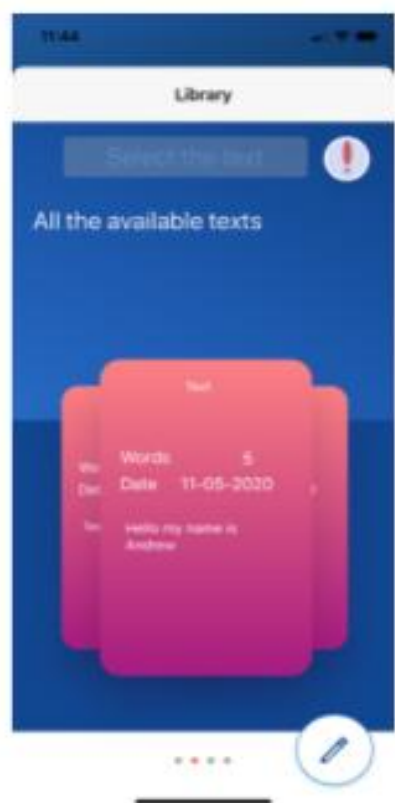
Нормоконтроль:

_____К.І.Ліщук

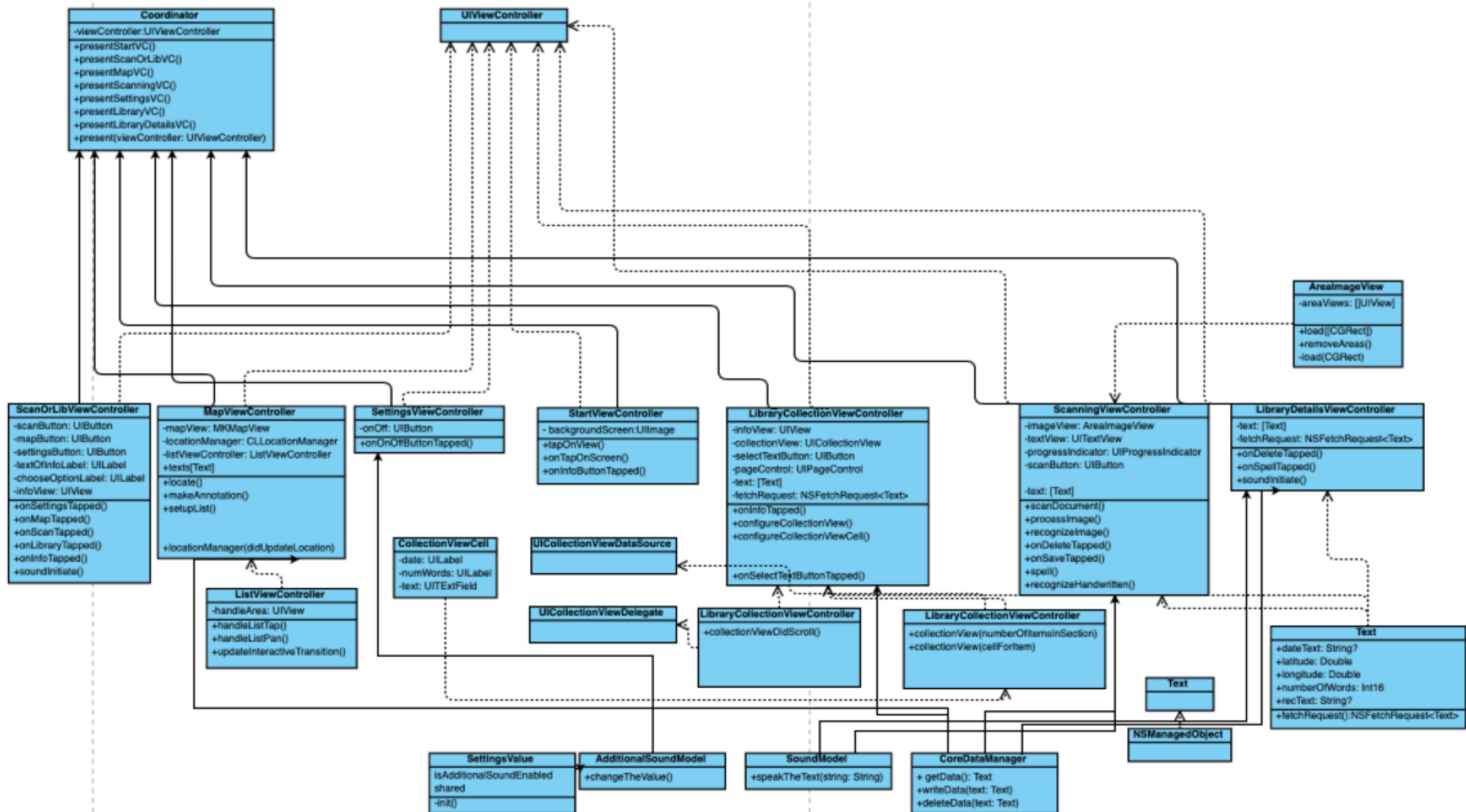
Виконавець:

_____А.Ю.Лісогор

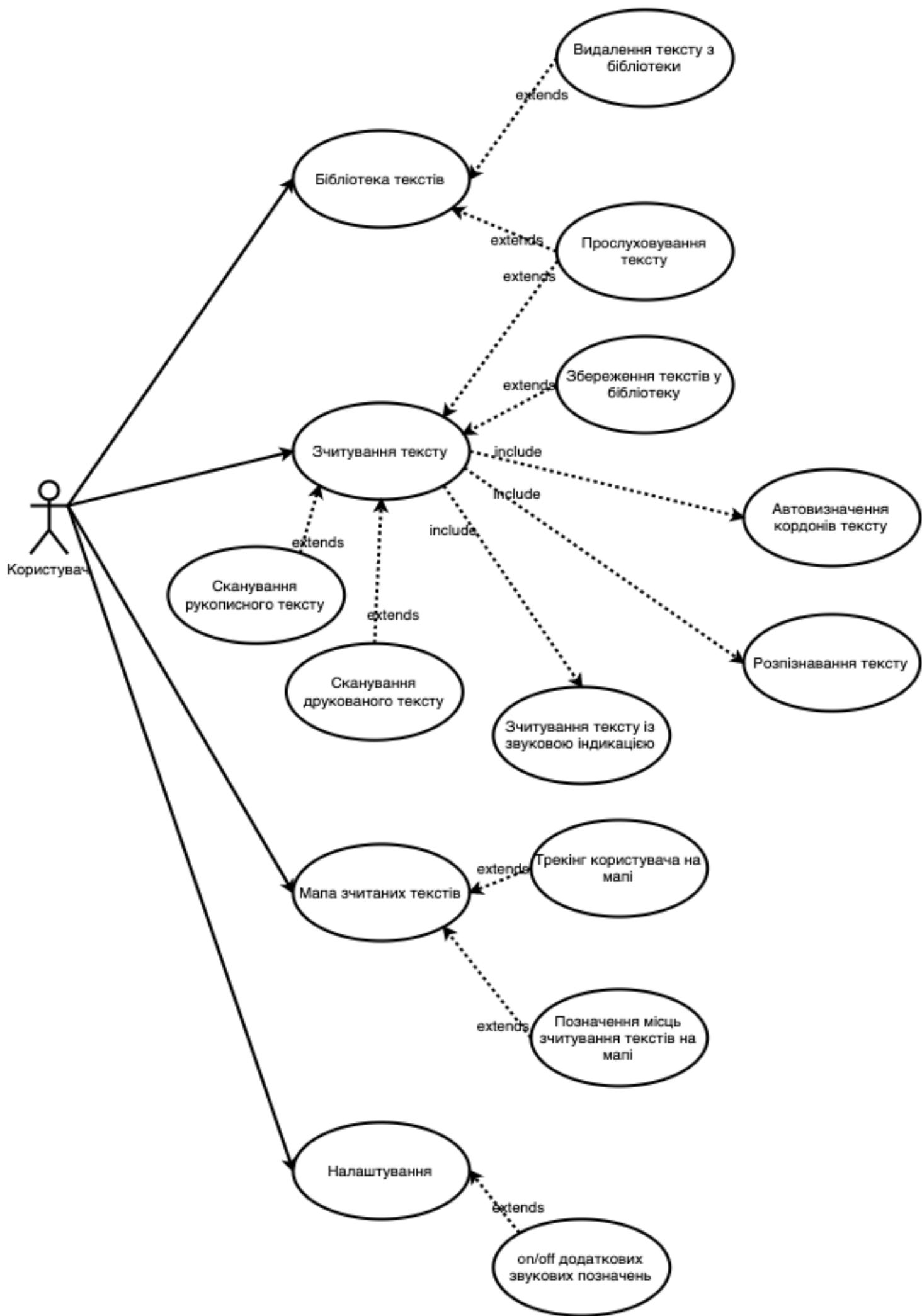
Київ – 2020 року



					КПІ.ІП-6317.045490.07.99.KE				
					Креслення вигляду екранних форм	Літера		Маса	Масштаб
Зробив	Арх.	№ документа	Підпис	Дата					
Перевіряв		Лісогор А.Ю.							
Затвердив		Халус О.А.							
Затвердив									
Затвердив					Мобільне застосування зі зчитування тексту для людей з вадами зору	Аркуш		Аркушів	
Затвердив									
Затвердив		Лісогор А.Ю.							
Затвердив		Халус О.А.			КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-63				



					КПІ.ІП-6317.045490.07.99.СС				
зм.	Арк.	№ документа	Підпис	Дата	Схема структурна класів програмного забезпечення	Літера		Маса	Масштаб
Розробив		Лісогор А.Ю.							
Перевірив		Халус О.А.							
з. кон.						Аркуш	Аркушів		
з. кон.		Лішук К.І.			Мобільне застосування зі зчитування тексту для людей з вадами зору	КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-63			
Затвердив		Халус О.А.							



					КПІ.ІП-6317.045490.07.99.СС					
					Схема структурна варіантів використання			Лит.	Арк.	Аркуші
Зм.	Арк.	№ докум.	Підп.	Дата	Мобільне застосування зі зчитування тексту для людей з вадами зору			Аркуш		Аркуші
Розроб.		Лісовор А.Ю.						1		
Перев.		Халус О.А.								
Т. Кон.										
					Мобільне застосування зі зчитування тексту для людей з вадами зору			КПІ ім.Ігоря Сікорського Кафедра АСОІУ		
Н. Кон.		Лісовор А.І.		гр. ІП-63						
Затв.		Халус О.А.								